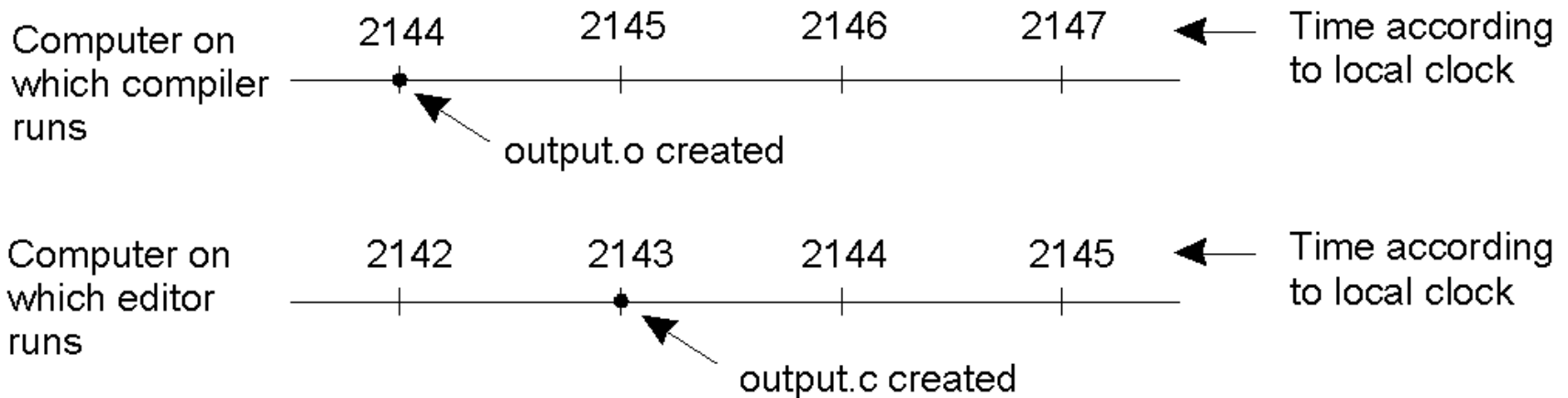


Szinkronizálás

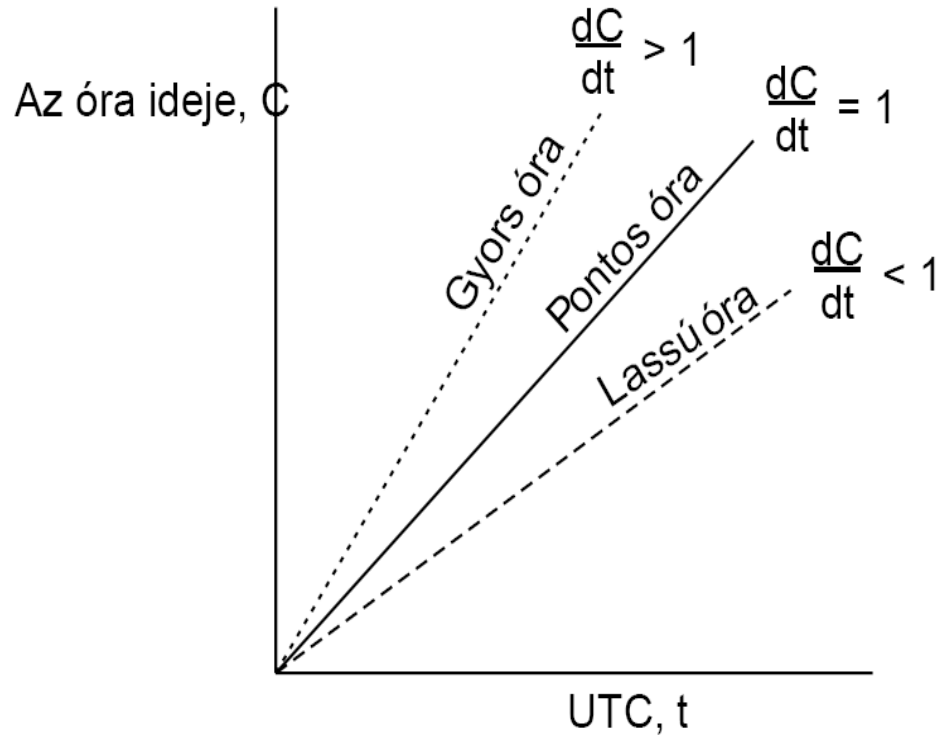
9. előadás

Az órák szinkronizálása



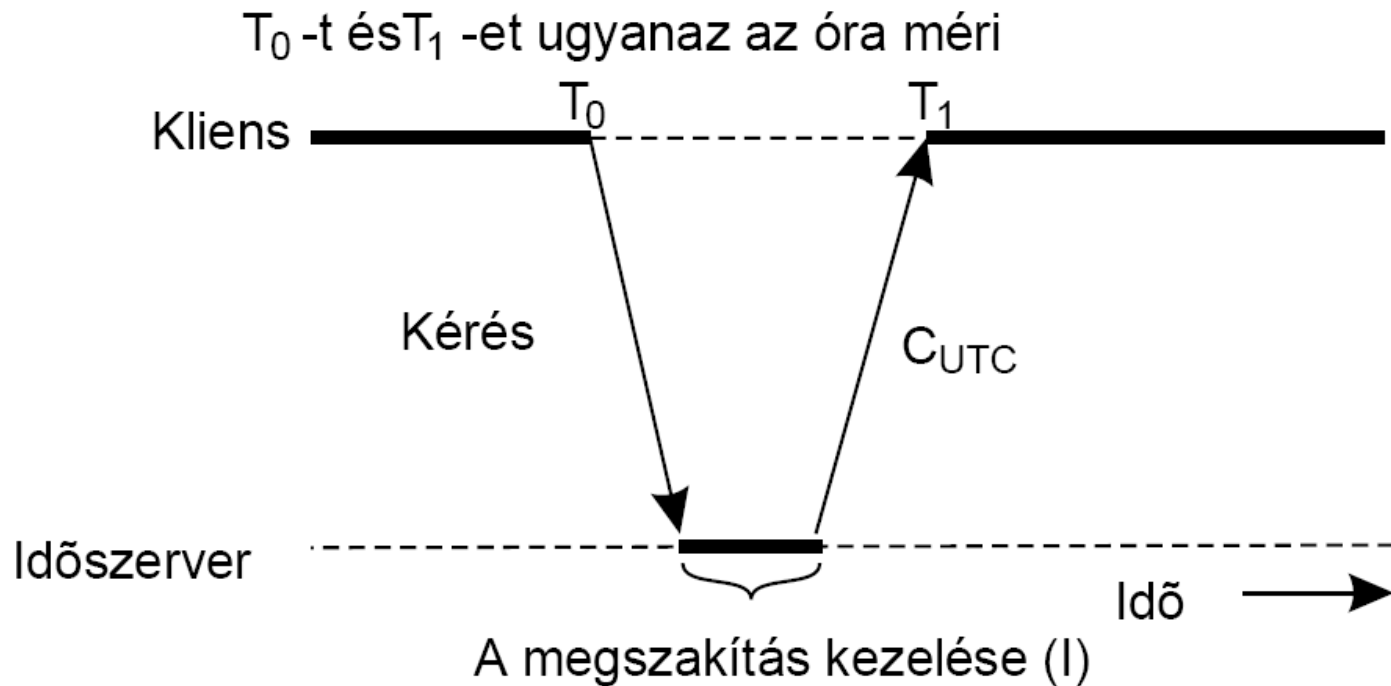
Ha mindegyik gép a saját óráját használja, akkor az adott esemény után történt másik eseményhez az elsőnél korábbi idő társulhat.

Óraszinkronizáló algoritmusok



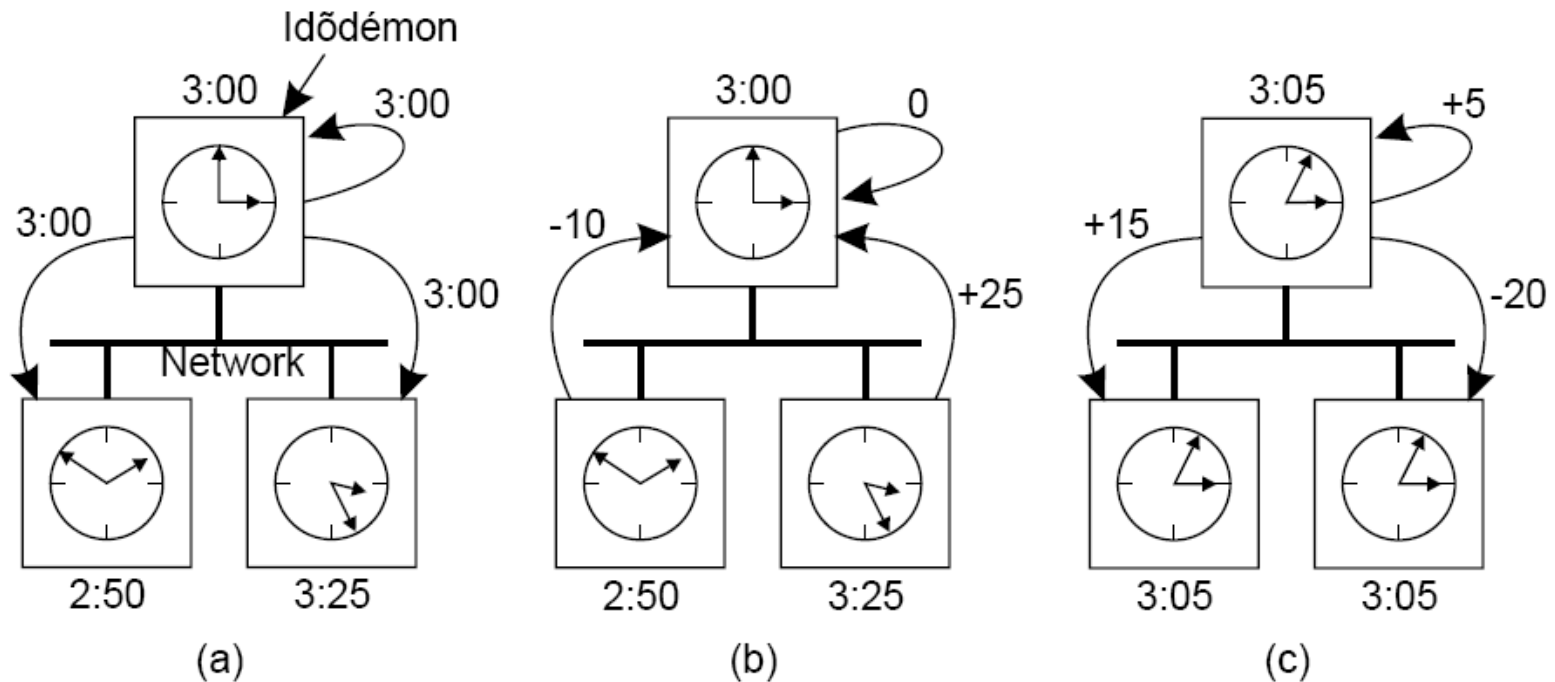
Az óra által mutatott idő és az UTC összefüggése az óra eltérése esetén.

Cristian algoritmusa



A pontos idő lekérdezése az időszerverről.

A Berkeley-algoritmus



- a) Az idődémon lekérdezi a többi géptől az általuk nyilvántartott pontos időt
- b) A gépek válaszolnak
- c) Az idődémon mindegyiket utasítja, hogyan állítsa be az óráját

Átlagoló algoritmusok

- Decentralizált algoritmus
- Újraszinkronizálási időszakok
- minden gép kihirdeti a saját idejét
- elindít egy időmérőt
- a beérkezett hírdetéseket összegyűjti

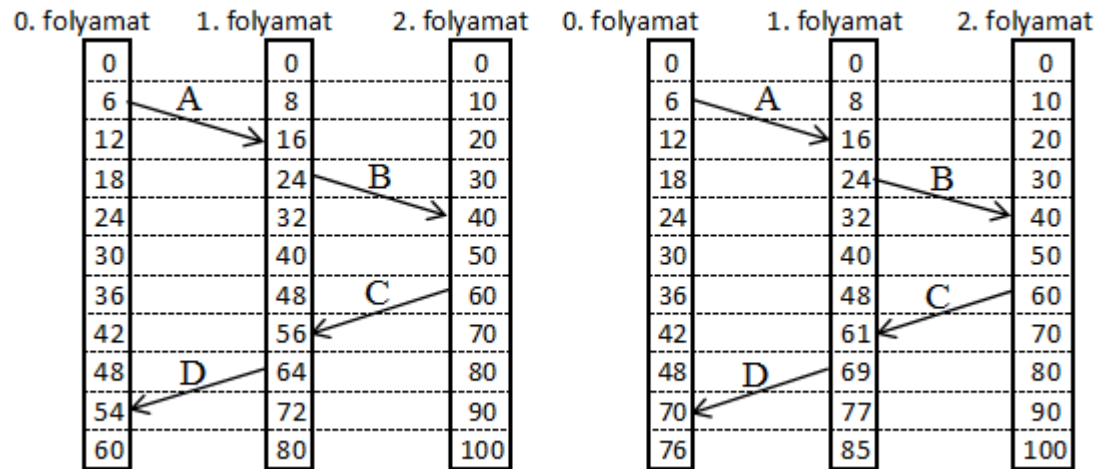
Többszörözött külső időforrások

- Több UTC-forrás vevő a rendszerben
- Ezek rendszeresen kihirdetik az idejüket

Logikai órák

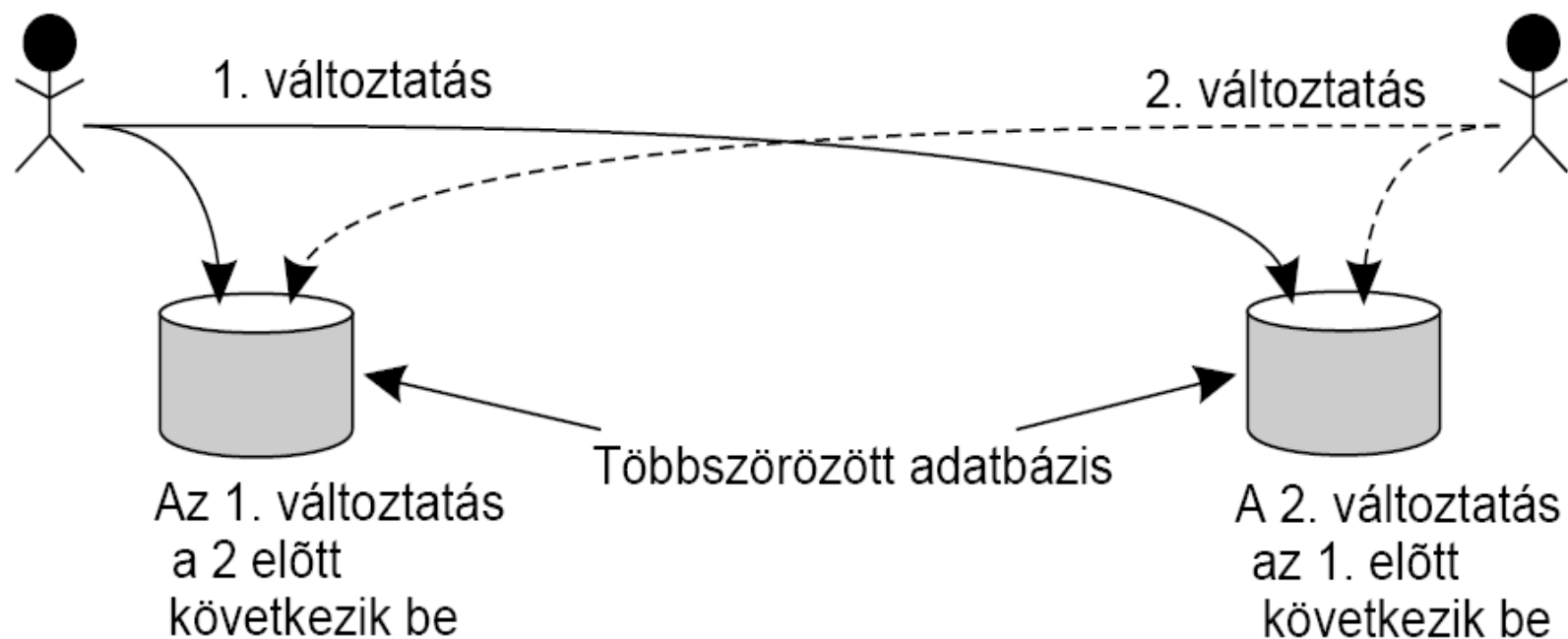
- Elegendő, ha az összes gép órája ugyanazt az időt mutatja
- Az események sorrendje a fontos

Lamport időbélyege



- Három folyamat, mindegyik saját órával. Az órák különböző sebességgel futnak.
- Lamport algoritmus az órák összehangolására.

Példa – pontosan sorbarendezett csoportcímkzés



A többszörözött adatbázis frissítése és az inkonzisztens állapot kialakulása.

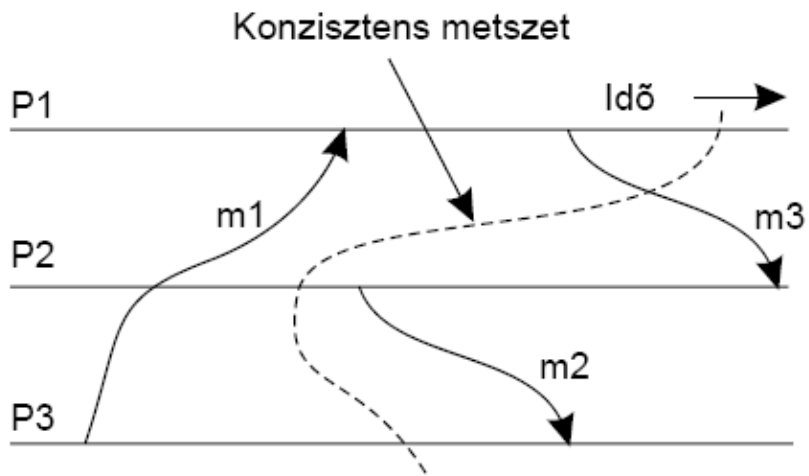
Időbélyeg vektor (1)

- ‘C(a)’ és ‘C(b)’ összehasonlítása semmit sem mond ‘a’ és ‘b’ kapcsolatáról
- Lamport időbélyegei nem foglalkoznak az oksági viszonyokkal
- Oksági viszony – időbélyeg-vektorok alkalmazása
- Ha ‘VT(a)’ < ‘VT(b)’ $\Rightarrow$ ‘a’ ‘b’ előzménye

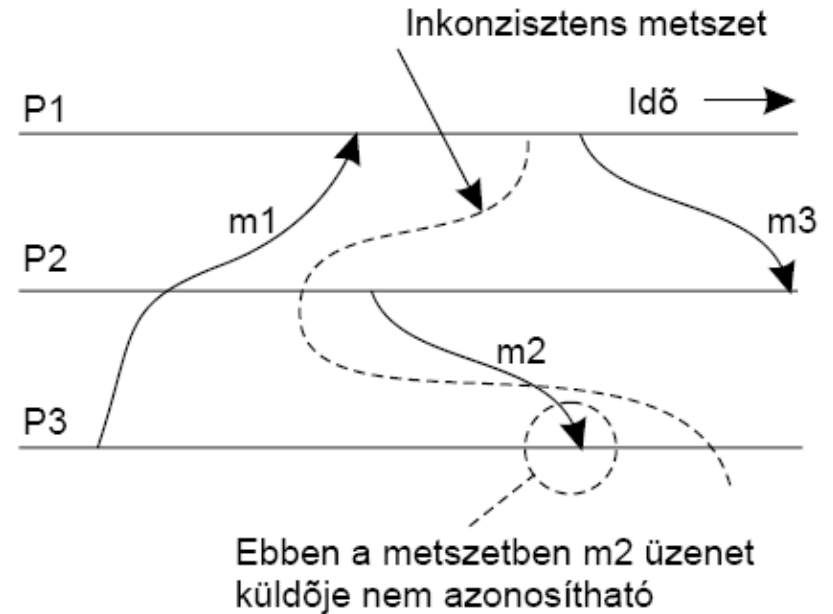
Időbélyeg vektor (2)

- P_i folyamat V_i vektorának tulajdonságai:
 1. $V_i[i]$ a P_i folyamatban eddig bekövetkezett események száma
 2. Ha $V_i[j] = k$, akkor P_i tudja, hogy a P_j folyamatban eddig k esemény történt
- Okozatilag összefüggő üzenetek küldése
- r csak akkor kézbesíti az üzenetet, ha:
 1. $vt(r)[j] = V_k[j] + 1$
 2. $vt(r)[i] < / = V_k[i]$ teljesül valamennyi $i < > j$

Globális állapot (1)



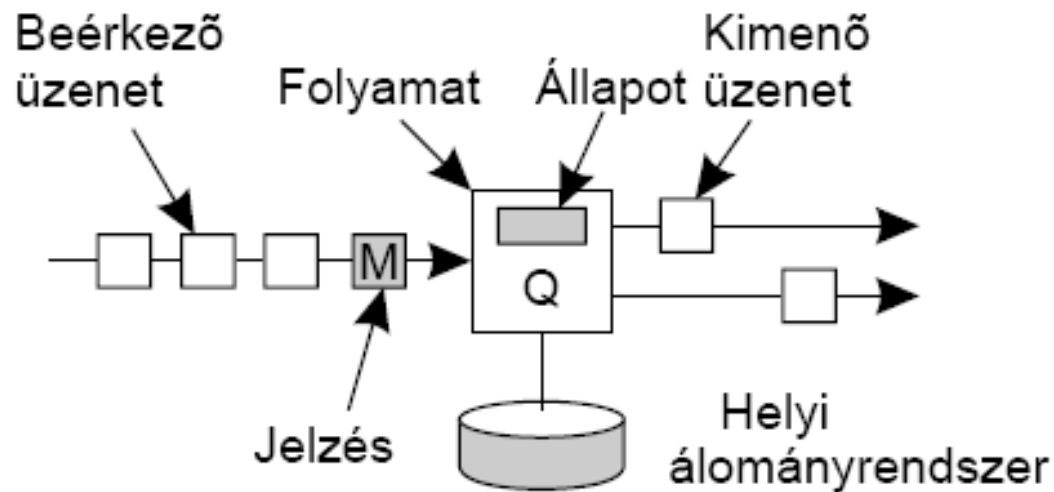
(a)



(b)

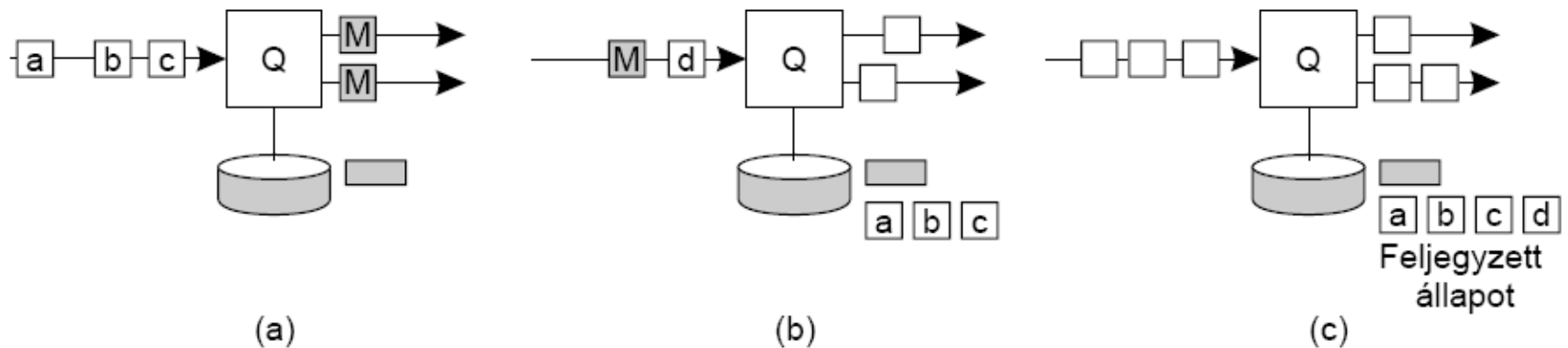
- a) Konzisztens metszet
- b) Inkonzisztens metszet

Globális állapot (2)



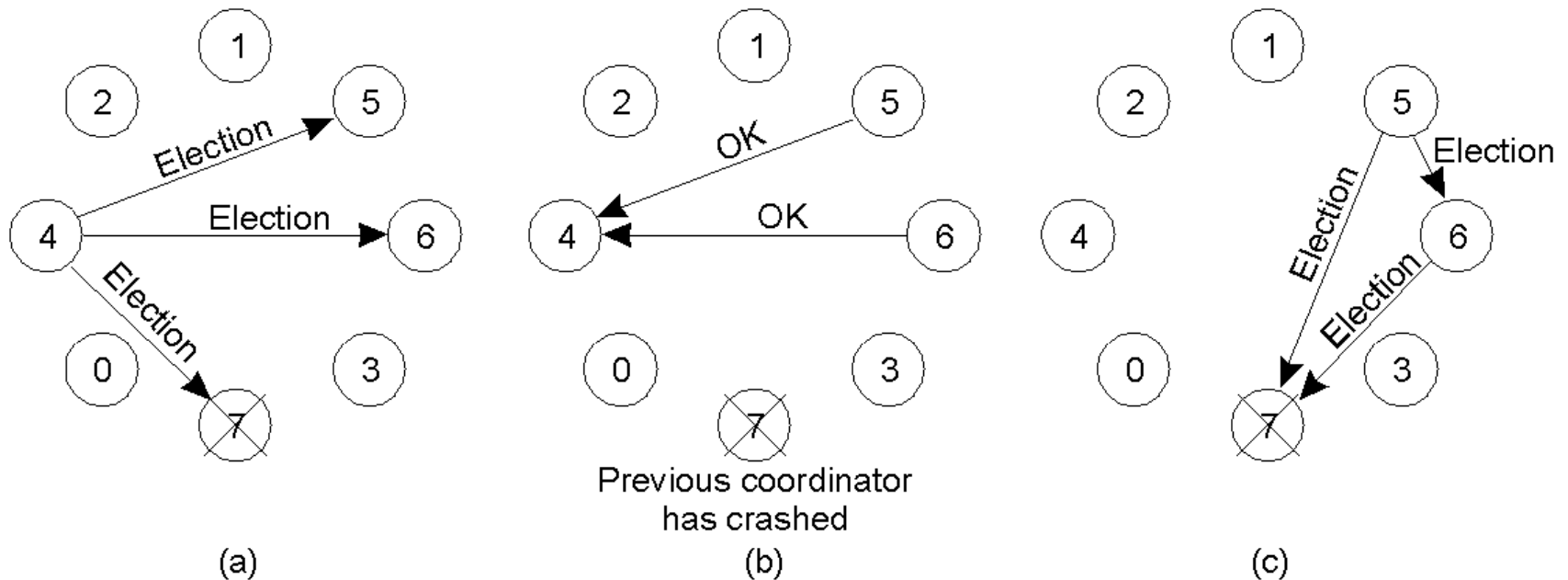
a) A folyamat és csatornáinak szerkezete az elosztott pillanatfelvétel elkészítéséhez.

Globális állapot (3)



- b) Q folyamat a jelzést először kapja meg, ezért feljegyzi saját állapotát.
- c) Q valamennyi érkező üzenetét feljegyzi.
- *d) Q megkapja a jelzést a bejövő csatornán és befejezi a csatorna*
- állapotának rögzítését.

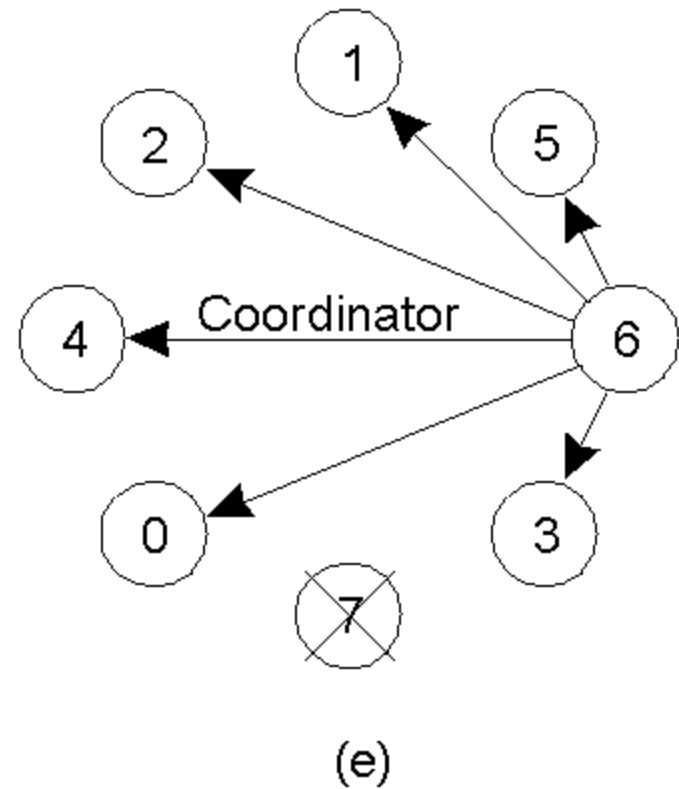
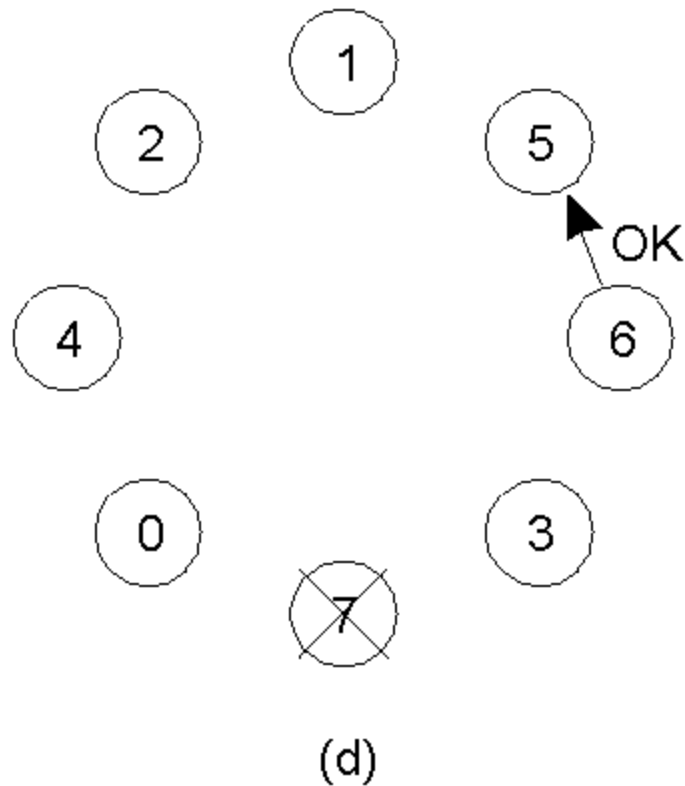
Szavazó algoritmusok - zsarnok (1)



A zsarnok algoritmus

- A 4-es folyamat választást tart
- 5-ös és 6-os válaszol, megmondva 4-esnek, hogy álljon le a választással
- 5-ös és 6-os választást kezdeményez

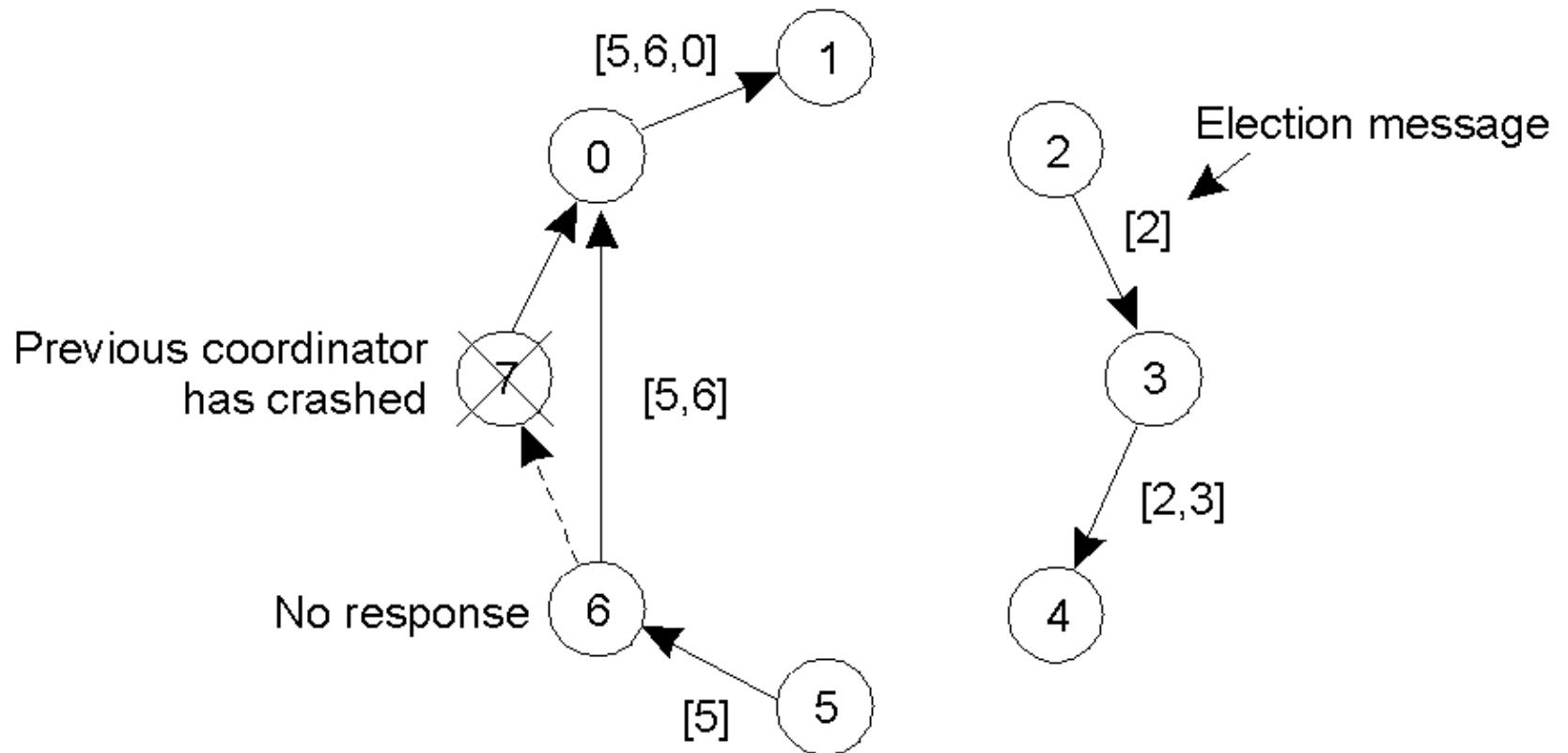
Szavazó algoritmusok - zsarnok (2)



d) 6-os szól 5-ösnek, hogy álljon le a választással

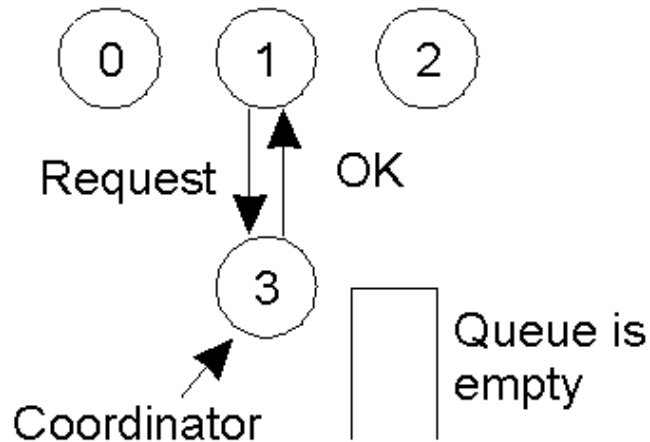
e) 6-os megnyeri a választást és kihirdeti a győzelmét

Szavazó algoritmusok - gyűrű

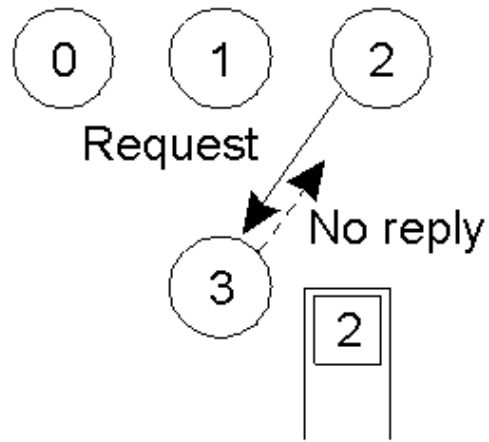


Kört használó szavazó algoritmus.

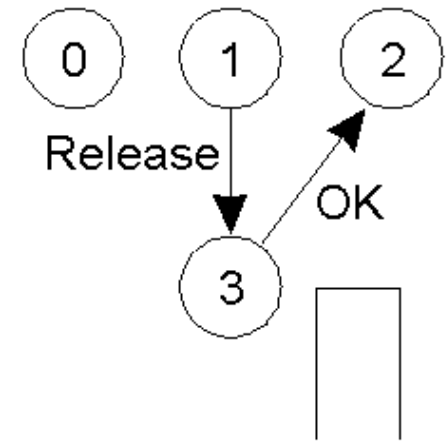
Kölcsönös kizárás: A központosított algoritmus



(a)



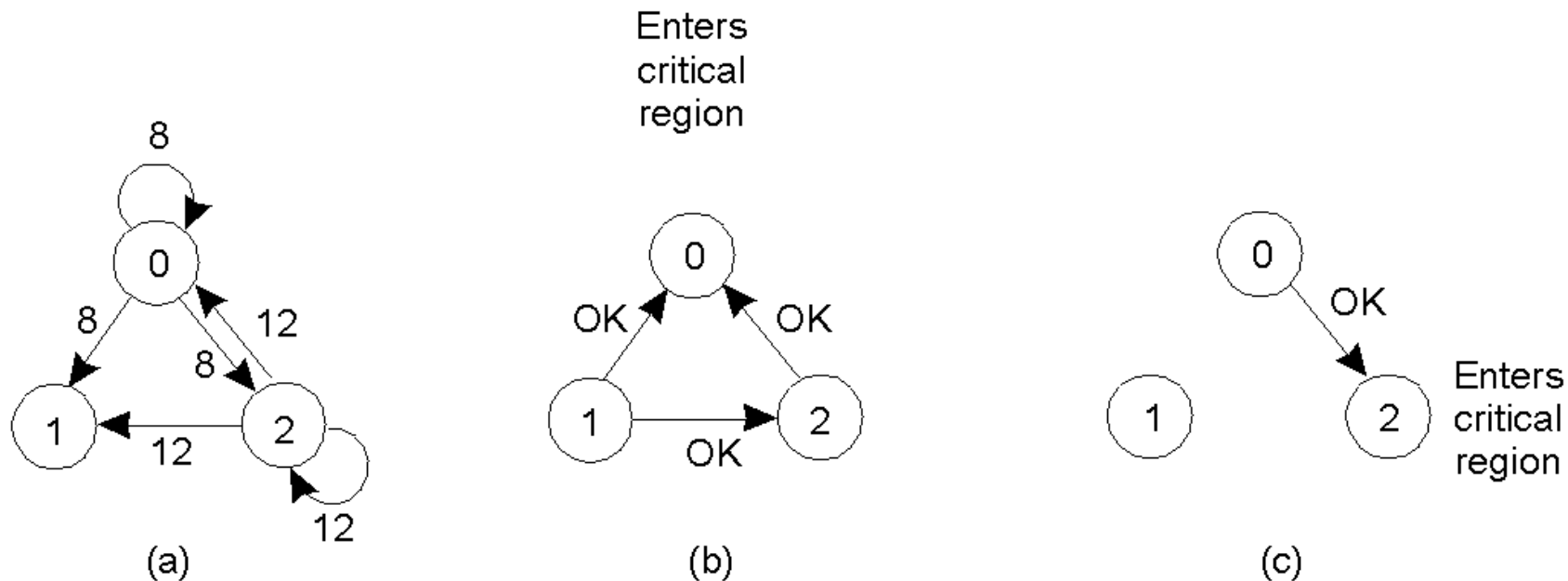
(b)



(c)

- a) Az 1-es folyamat engedélyt kér a koordinátortól, hogy beléphessen a kritikus területre. Az engedély megadva.
- b) A 2-es folyamat engedélyt kér ugyanerre a kritikus területre való belépéshez. A koordinátor nem válaszol.
- c) Amikor az 1-es folyamat elhagyja a kritikus területet, értesíti erről a koordinátort, aki ezután válaszol a 2-es folyamatnak.

Kölcsönös kizárás: Az elosztott algoritmus

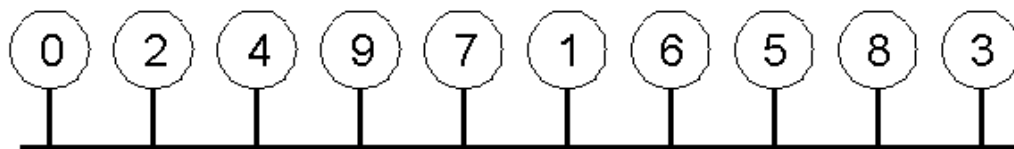


a) Két folyamat akar ugyanabban a pillanatban belépni ugyanarra a kritikus területre.

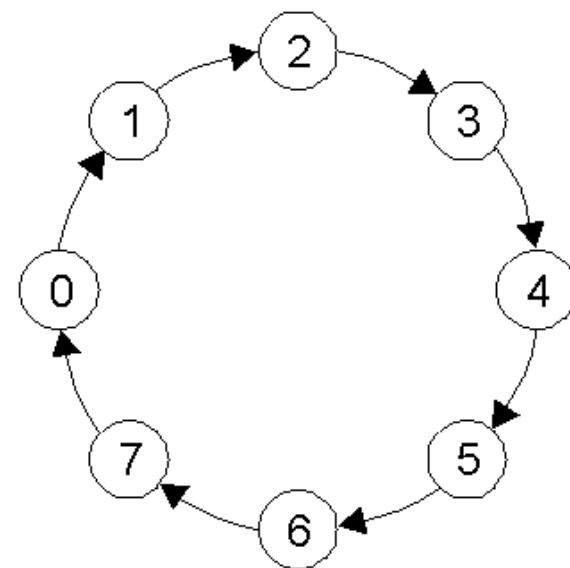
b) A 0-ás folyamatnak alacsonyabb az időbélyege, így ő nyer.

c) Amikor a 0-ás folyamat végzett, ő is elküldi az OK választ, így a 2-es folyamat már beléphet a kritikus területre.

Kölcsönös kizárás: A zsetongyűrű algoritmus



(a)



(b)

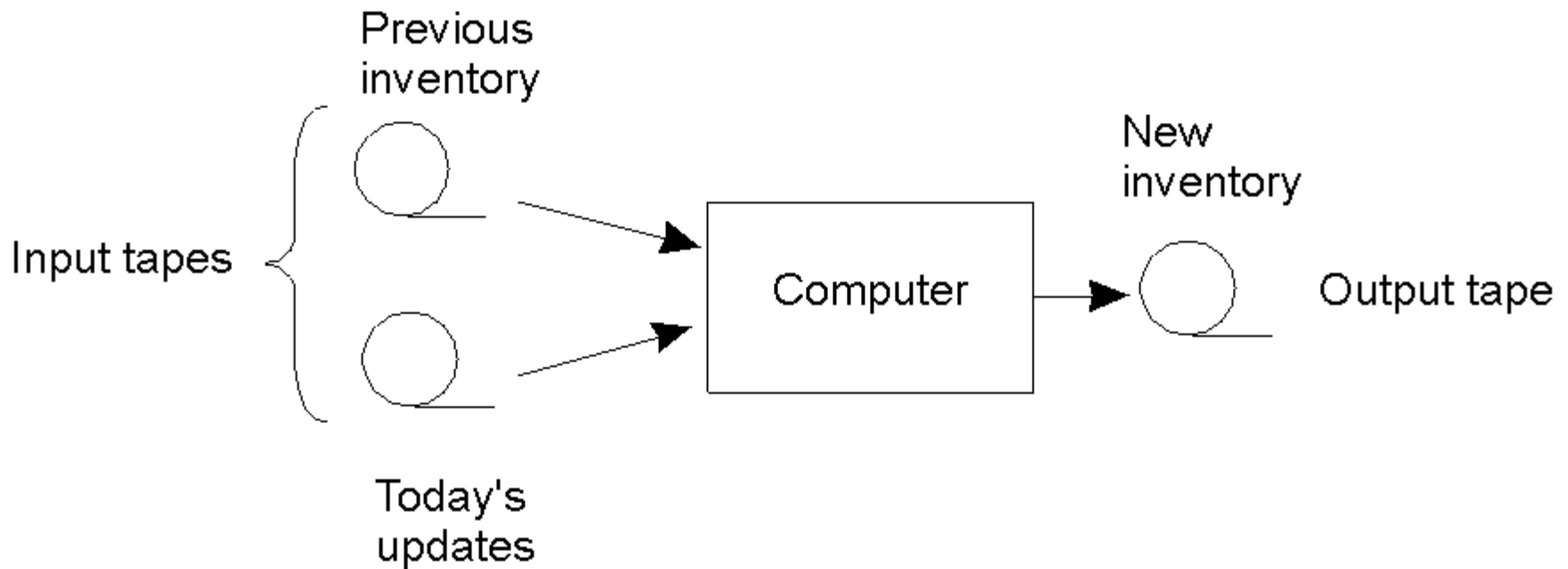
- a) A folyamatok rendezetlen csoportja a hálózaton.
- b) A szoftver által alkotott logikai gyűrű.

Kölcsönös kizárás: összehasonlítás

Algoritmus	Belépési / kilépési üzenetszám	Belépés előtti készletelés (üzenetidőben)	Problémák
Központosított	3	2	A koordinátor összeomlása
Elosztott	$2 (n - 1)$	$2 (n - 1)$	Bármely folyamat összeomlása
Zsetongyűrű	1 to ∞	0 to $n - 1$	Elveszett zseton, a folyamat összeomlása

A kölcsönös kizárást biztosító algoritmusok
összehasonlítása.

Elosztott tranzakciók - tranzakciómodell (1)



A mesterszalag frissítése hibatűrő folyamat.

Elosztott tranzakciók - tranzakciómodell (2)

Primitív	Leírása
BEGIN_TRANSACTION	Tranzakció kezdetének jelzése
END_TRANSACTION	Tranzakció végének jelzése
ABORT_TRANSACTION	A tranzakció félbeszakítása és az eredeti állapot visszaállítása
READ	Adat beolvasása állományból, táblából vagy bárhol
WRITE	Adat kiírása állományba, táblába vagy bárhová

Néhány példa a tranzakció-primitívekre.

Elosztott tranzakciók - tranzakciómodell (3)

```
BEGIN_TRANSACTION  
  reserve WP -> JFK;  
  reserve JFK -> Nairobi;  
  reserve Nairobi -> Malindi;  
END_TRANSACTION
```

(a)

```
BEGIN_TRANSACTION  
  reserve WP -> JFK;  
  reserve JFK -> Nairobi;  
  reserve Nairobi -> Malindi full =>  
ABORT_TRANSACTION
```

(b)

a) 4 járat helyfoglalása egy tranzakcióként

b) A tranzakció félbeszakad, amikor a negyedik járat nem elérhető

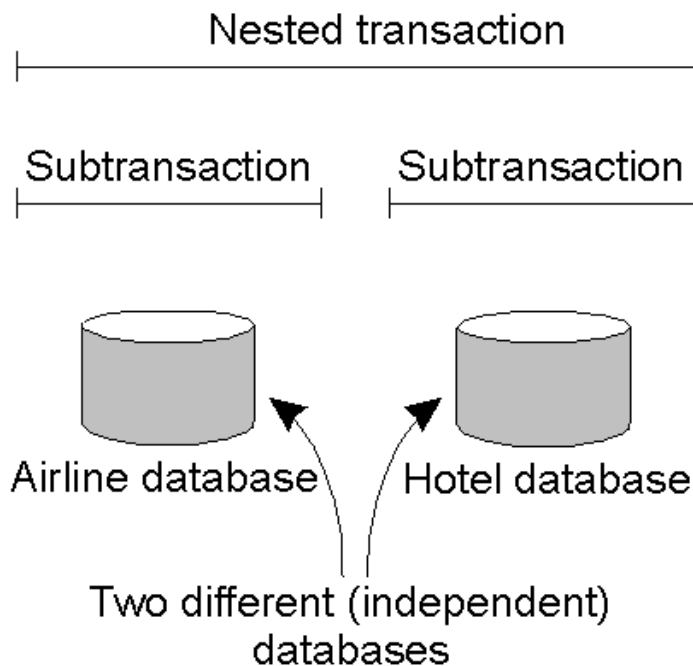
A tranzakciók tulajdonságai (ACID)

1. Oszthatatlan (elemi) – Atomic
2. Konzisztens – Consistent
3. Elkülönülő – Isolated
4. Maradandó - Durable

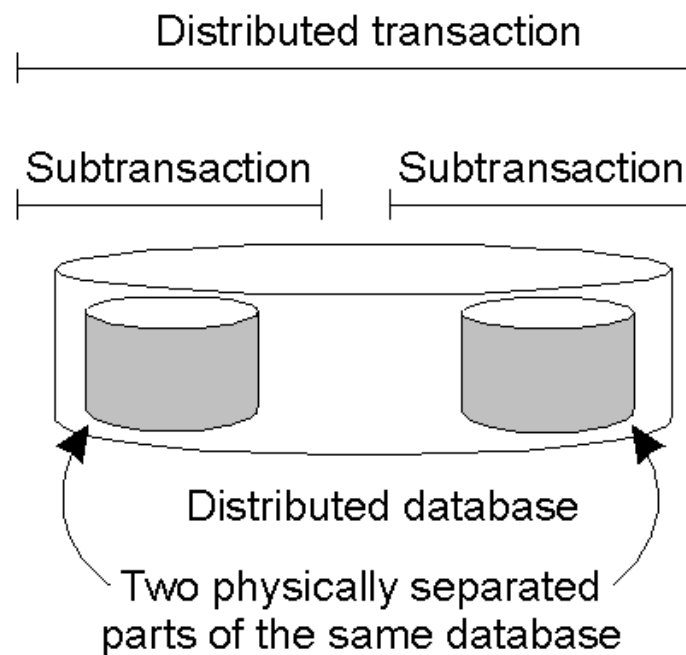
A tranzakciók osztályozása

- Az egységes tranzakciók korlátai
 - Beágyazott tranzakciók
 - Elosztott tranzakciók

Elosztott tranzakciók



(a)

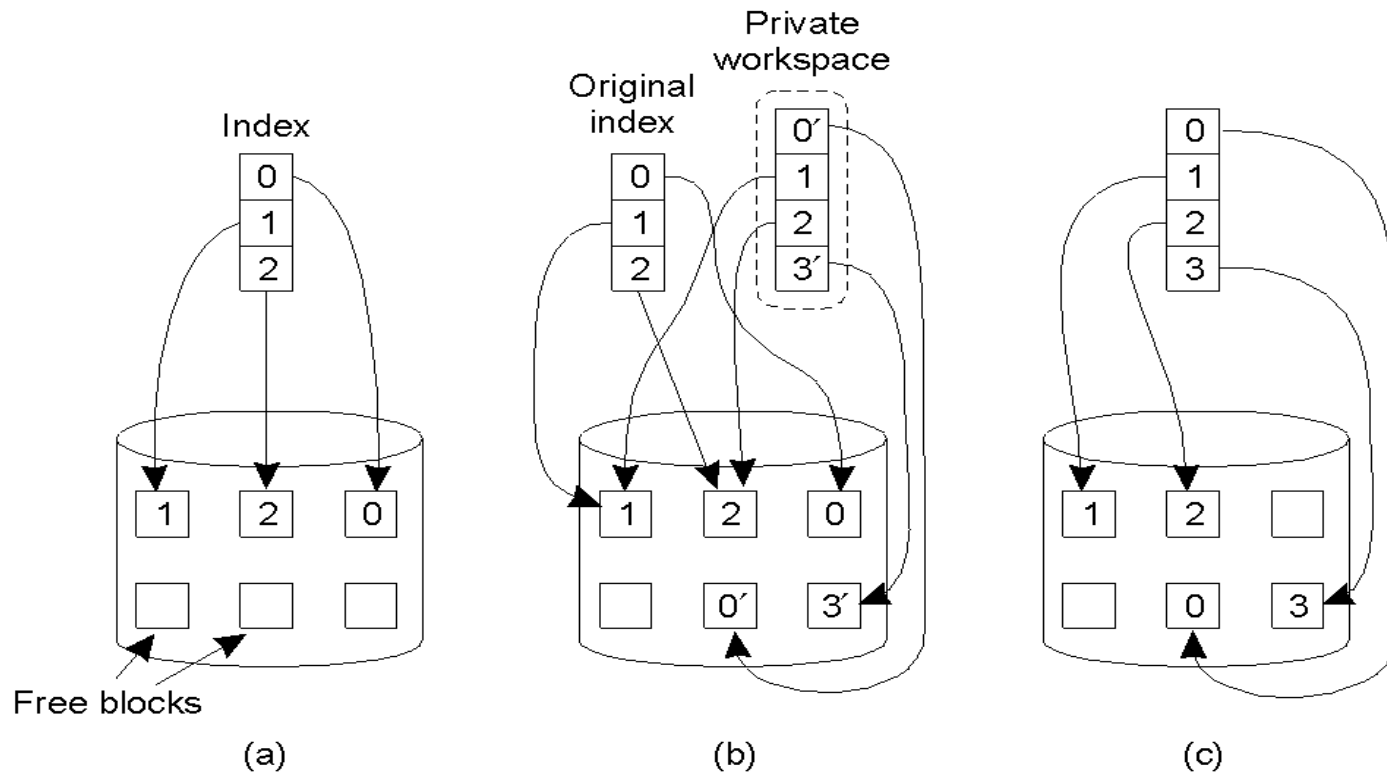


(b)

a) Beágyazott tranzakció

b) Elosztott tranzakció

Implementáció – saját munkaterület



- a) Egy három blokk hosszúságú állomány indexei és lemezblokkjai
- b) Az állomány 0-ás blokkjának módosítása a tranzakció során, és egy negyedik blokk hozzáfűzése után kialakult helyzet
- c) A tranzakció lezárása utáni helyzet

Implementáció – munkatervnapló

x = 0;	Log	Log	Log
y = 0;			
BEGIN_TRANSACTION;			
x = x + 1;	[x = 0 / 1]	[x = 0 / 1]	[x = 0 / 1]
y = y + 2		[y = 0/2]	[y = 0/2]
x = y * y;			[x = 1/4]
END_TRANSACTION;			
(a)	(b)	(c)	(d)

a) Egy tranzakció

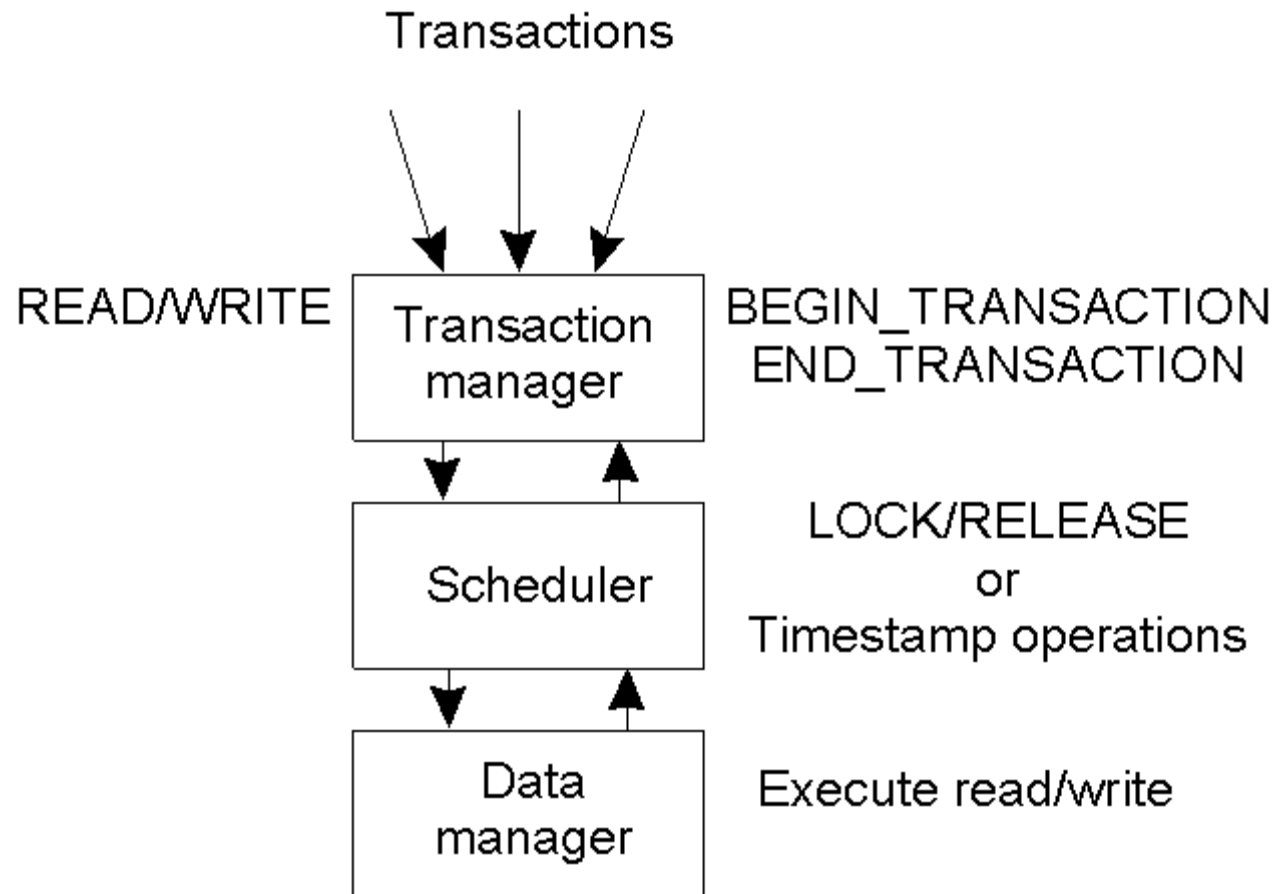
b) – d) Az egyes utasítások végrehajtása előtt kiírt naplóbejegyzés

Konkurenciakontroll

A tranzakciók tulajdonságai (ACID)

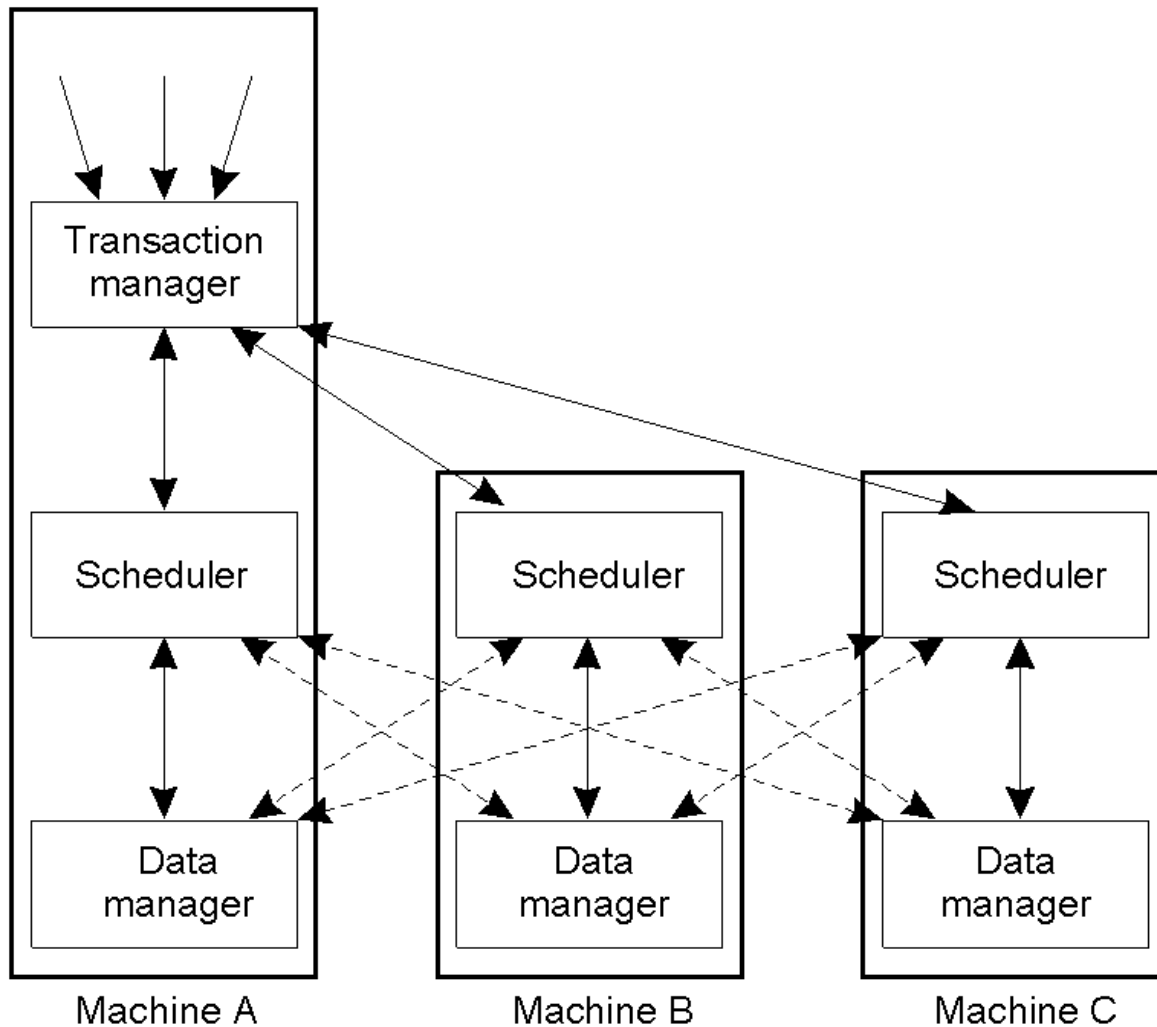
- 1. Oszthatatlan (elemi) - Atomic
- 2. Konzisztens - Consistent
- **3. Elkülönülő – Isolated**
- [sorosítható – serializable]
- 4. Maradandó - Durable

Konkurenciakontroll (1)



- A tranzakciókat megvalósító kezelők általános felépítése.

Konkurenciakontroll (2)



Sorosíthatóság (1)

BEGIN_TRANSACTION
x = 0;
x = x + 1;
END_TRANSACTION

(a)

BEGIN_TRANSACTION
x = 0;
x = x + 2;
END_TRANSACTION

(b)

BEGIN_TRANSACTION
x = 0;
x = x + 3;
END_TRANSACTION

(c)

Schedule 1	x = 0; x = x + 1; x = 0; x = x + 2; x = 0; x = x + 3	Legal
Schedule 2	x = 0; x = 0; x = x + 1; x = x + 2; x = 0; x = x + 3;	Legal
Schedule 3	x = 0; x = 0; x = x + 1; x = 0; x = x + 2; x = x + 3;	Illegal

(d)

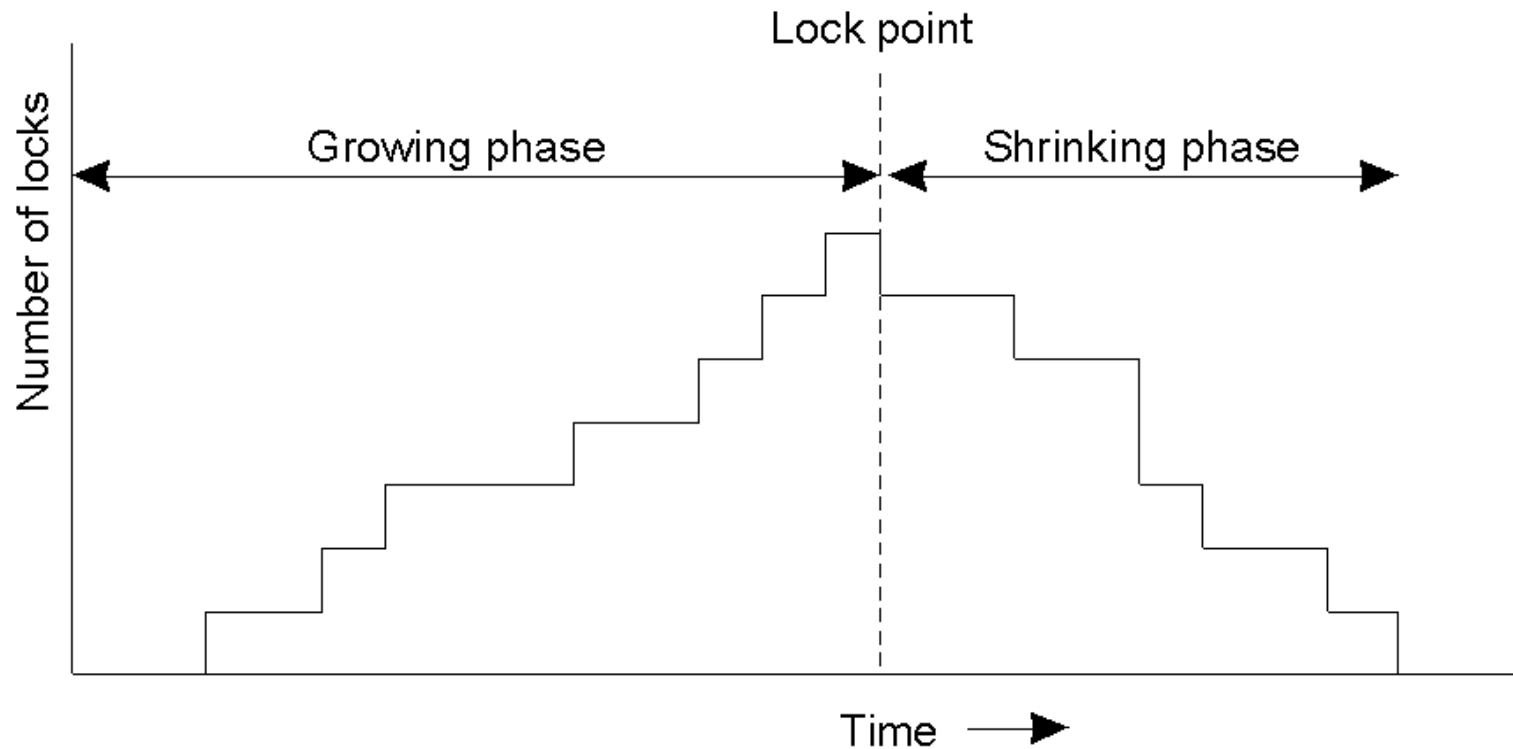
a) – c) Három tranzakció: T1, T2, and T3

d) Lehetséges ütemtervek

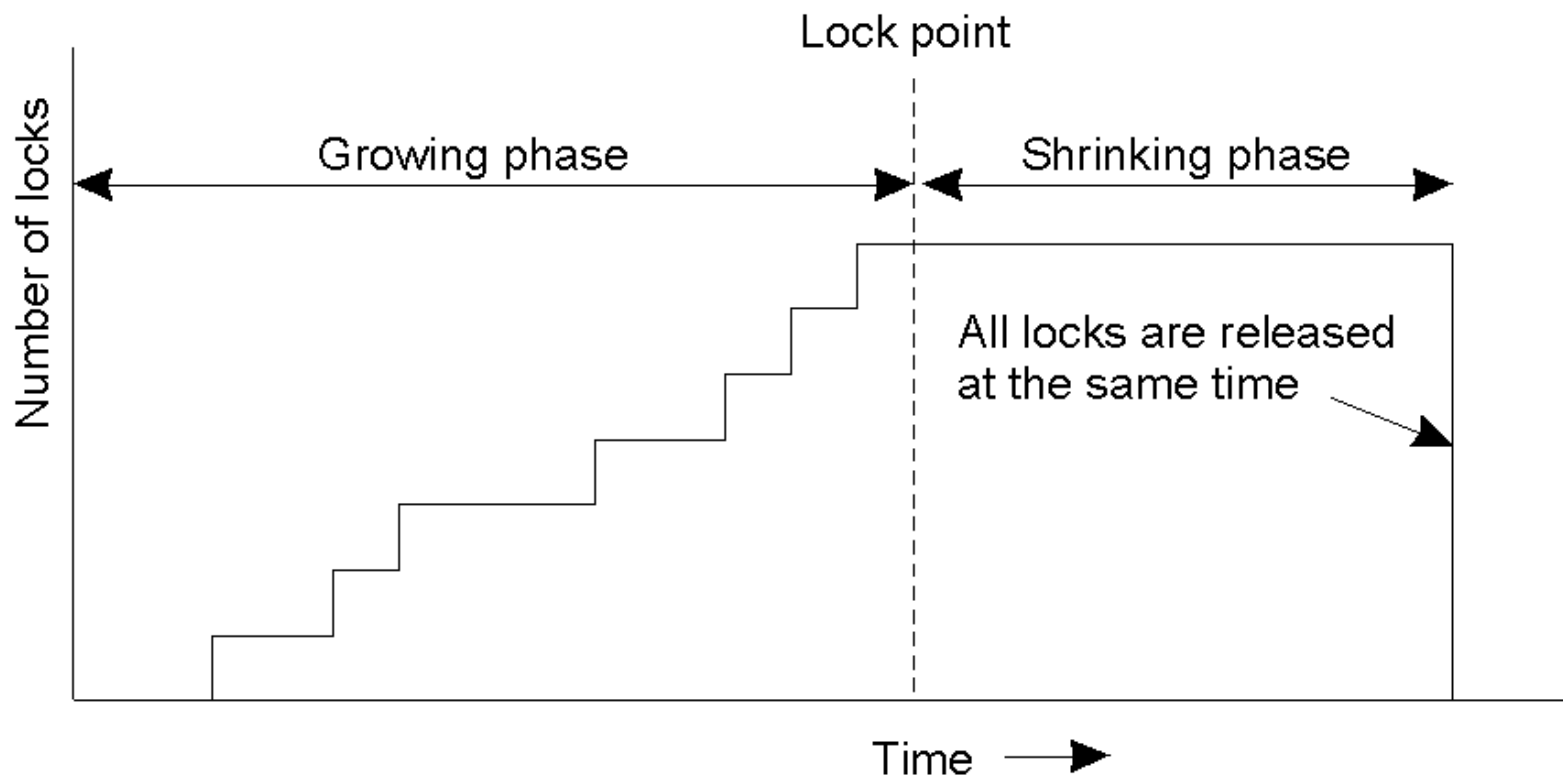
Sorosíthatóság (2)

- `write(Ti, x); read(Ti, x); write(Ti, x)`
- ütköző műveletek
- Szinkronizálás
- kölcsönös kizárás
- Időbélyegek
- Megközelítés
- Pesszimista
- optimista

Kétfázisú zárolás (1)



Kétfázisú zárolás (2)

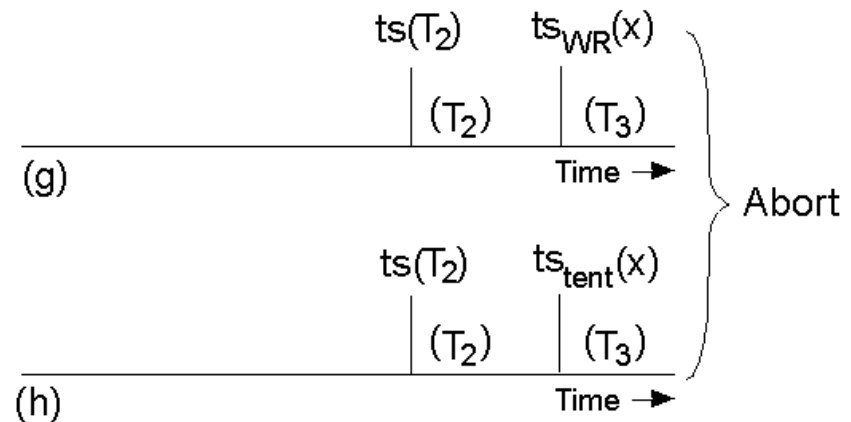
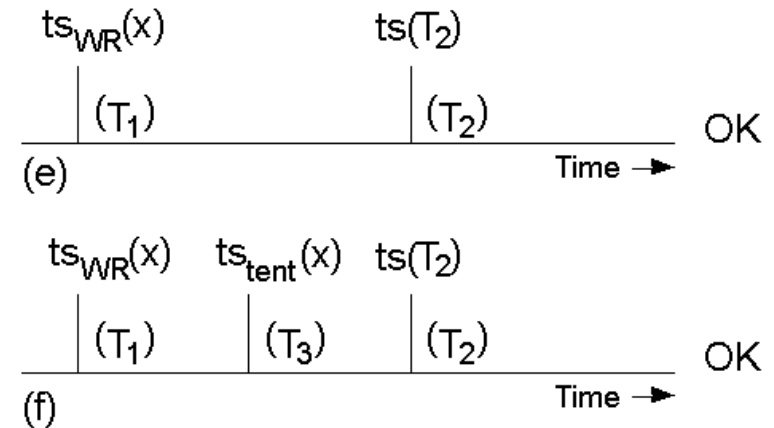
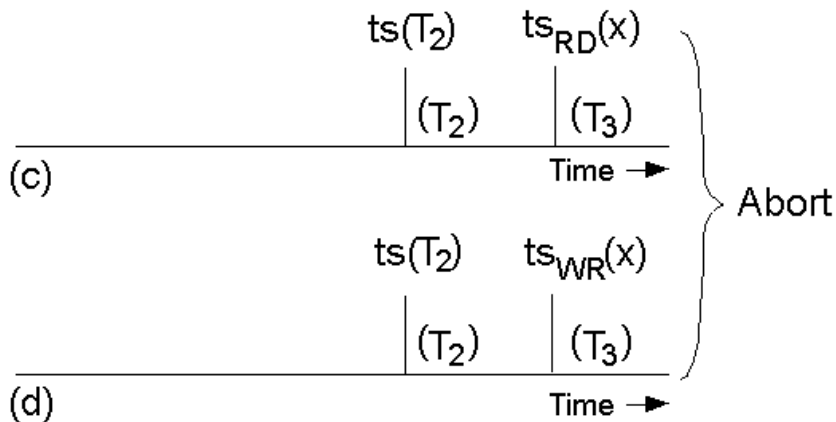
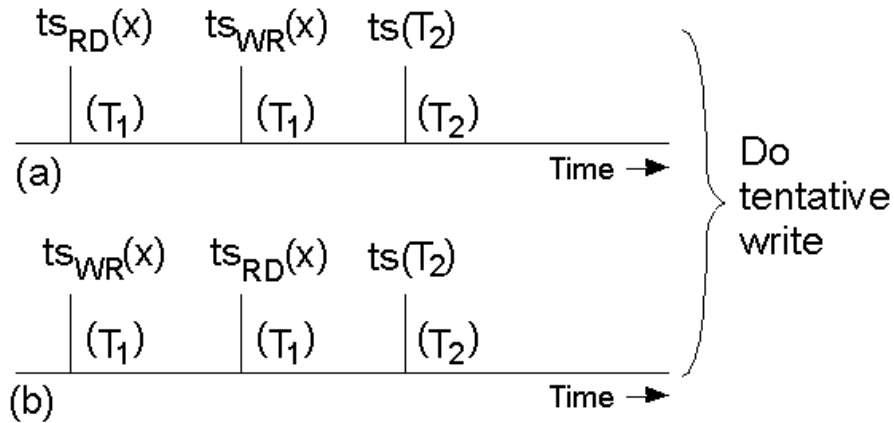


Szigorú kétfázisú zárolás

Pesszimista időbélyeg-rendezés (1)

- T tranzakció – $ts(T)$ kezdeti időbélyeg
- x adatalem – $tsRD(x)$, $tsWR(x)$ utolsó olvasást / írást végző tranzakció időbélyege
- $read(T, x)$, ahol $ts(T) < tsWR(x) \Rightarrow T$ megszakad
- $read(T, x)$, ahol $ts(T) > tsWR(x) \Rightarrow OK$
- $[tsRD(x) \leftarrow \max\{ts(T), tsRD(x)\}]$
- $write(T, x)$, ahol $ts(T) < tsRD(x) \Rightarrow T$ megszakad
- $write(T, x)$, ahol $ts(T) > tsRD(x) \Rightarrow OK$
- $[tsWR(x) \leftarrow \max\{ts(T), tsWR(x)\}]$

Pesszimista időbélveg-rendezés (2)



Az időbélyeget használó konkurenciakontrol.

Optimista időbélyeg-rendezés

- Saját munkaterület
- Ha az ütközések ritkák tranzakció közben nincs vizsgálat, csak nyilvántartás (írások, olvasások)
- Tranzakció lezárásakor az összes többi tranzakció vizsgálata megszakad, vagy lezárul
- Időbélyeg használata