



TD 03 - *Gestion des transactions*

Exercice 01

1) Construisez les graphes de sérialisation pour les trois exécutions (histoires) suivantes. Indiquez les exécutions sérialisables et vérifiez si parmi les trois exécutions il y a des exécutions équivalentes.

1. **H1: w2[x] w3[z] w2[y] c2 r1[x] w1[z] c1 r3[y] c3**

2. **H2: r1[x] w2[y] r3[y] w3[z] c3 w1[z] c1 w2[x] c2**

3. **H3: w3[z] w1[z] w2[y] w2[x] c2 r3[y] c3 r1[x] c1**

2) Indiquez l'ordre d'exécution établi par l'ordonnanceur, en considérant que les opérations bloquées en attente d'un verrou seront exécutées en priorité dès que le verrou devient disponible, dans l'ordre de leur blocage. On suppose que les verrous d'une transaction sont relâchés au moment du Commit.

H : r1[x] r2[y] w3[x] w 1[y] w1[x] w2[y] c2r3[y] r1[y] c1 w3[y] c3

Exercice 02

Les trois programmes suivants peuvent s'exécuter dans un système de gestion bancaire. **Débit** diminue le solde d'un compte (c) avec un montant donné (m). Pour simplifier, tout debit est permis (on accepte des découverts). **Crédit** augmente le solde d'un compte (c) avec un montant donné (m). **Transfert** transfère un montant (m) à partir d'un compte source (s) vers un compte destination (d). L'exécution de chaque programme démarre par un **Start** et se termine par un **Commit** (non montrés ci-dessous).

Débit (c:Compte ; m:Montant)	Crédit (c:Compte; m:Montant)	Transfert (s,d:Compte; m:Montant)
Begin	begin	Begin
t := Read(c);	t = Read(c);	Débit(s,m);
Write(c,t-m);	Write(c,t+m) ;	Crédit(d,m);
End	End	end

Le système exécute en même temps les trois opérations suivantes: (1) un transfert de montant 100 du compte A vers le compte B, (2) un crédit de 200 pour le compte A, (3) un débit de 50 pour le compte B.

1. Écrire les transactions T_1 , T_2 et T_3 qui correspondent à ces opérations. Montrer que l'histoire :

H: r1[A] r3[B] w1[A] r2[A] w3[B] r1[B] c3 w2[A] c2 w1[B] c1 est une exécution concurrente de T_1 , T_2 et T_3 .

2. Mettre en évidence les conflits dans **H** et construire le graphe de sérialisation de cette histoire. **H** est-elle sérialisable?

3. Quelle est l'exécution **H'** obtenue à partir de **H** par verrouillage à deux phases ? On suppose que les verrous d'une transaction sont relâchés après le Commit de celle-ci. Une opération bloquée en attente d'un verrou bloque le reste de sa transaction. Au moment du relâchement des verrous, les opérations en attente sont exécutées en priorité.

Si au début le compte A avait un solde de 100 et B de 50, quel sera le solde des deux comptes après la reprise si une panne intervient après l'exécution de $w_1[B]$?

Exercice 03

On considère les séquences d'opérations suivantes ou **st**, signifie que la transaction i commence :

S0 : st1;st2;st3;st4;r1(A);w1(A);w4(A);r3(A);w3(A);w2(A);r2(A);st5;st6;st7;w5(A);r2(A);r7(A);r6(A)

S1 : st1;st2;st3;st4;w1(A);w3(A);r4(A);r2(A);

S2 : st1;st2;st3;st4;w1(A);w4(A);r3(A);w2(A);

S3 : st1;st2;st3;st4;w1(A);w2(A);w3(A);r2(A);r4(A);

1. Décrire l'exécution de chacune des suites d'opérations avec :

- un système d'estampillage simple.
- un système d'estampillage multiple

Exercice 04

Considérons une relation **Compte** (Numéro, Titulaire, Solde) et trois opérations qui s'exécutent sur cette relation à deux niveaux de granularité : **relation et enregistrement**. Supposons que dans cette relation il y a deux comptes pour le titulaire "Omar".

Les trois opérations sont les suivantes :

- ✓ un programme qui lit tous les comptes ;
- ✓ un programme qui réalise un transfert entre les deux comptes de "Omar" ;
- ✓ un programme qui lit tous les comptes, afin de trouver les comptes de "Omar" et de créditer leur solde avec les intérêts de l'année.

1. Montrer que l'exécution suivante est une exécution concurrente des trois opérations ci-dessus. Identifier les données qui correspondent à chaque variable.

H : r1[z] r2[z] w1[x] r3[x] w3[x] c2 w1[y] r3[y] c1 w3[y] c3

2. Trouver l'exécution obtenue par verrouillage hiérarchique à partir de H. On considère que le verrouillage hiérarchique utilise des verrous de type **SIX** (lecture et intention d'écriture), dans le cas présent pour le rajout des intérêts sur les comptes de "Omar". Les verrous sont relâchés au Commit et les opérations bloquées en attente de verrou sont traitées en priorité, dans l'ordre de leur blocage.

3. Que se passe-t-il dans le verrouillage hiérarchique de H si on n'utilise pas de verrou SIX au niveau de la relation (mais seulement du S et du IX séparément) ?

Exercice 05

L'histoire suivante est une exécution concurrente de trois transactions dans un système de location de DVD.

$H : r_1[x]r_2[y] w_1[y] r_3[y] w_1[z] w_2[y] c_1 w_3[z] c_3 c_2$

1. L'un des programmes de ce système réalise la modification du prix de location d'un DVD en cas de promotion. Ce programme exécute les opérations suivantes: il demande à l'administrateur du système d'introduire le code du DVD et le taux de réduction du prix et diminue dans la base de données le prix avec le pourcentage donné. **Laquelle des trois transactions de H représente l'exécution de ce programme? Pourquoi certaines opérations du programme n'apparaissent pas dans la transaction? Justifiez votre réponse.**
2. Identifiez tous les conflits dans H et vérifiez si l'exécution est sérialisable en construisant le graphe de sérialisation.
3. Quelle est l'exécution obtenue par verrouillage à deux phases à partir de H?
4. Notons H' l'histoire obtenue suite au verrouillage à deux phases. Peut-on dire, sans construire le graphe de sérialisation, que H' est sérialisable ? Pourquoi?