

Examen NMV
M2 – MASTER SAR
Novembre 2009

2 heures – Tout document papier autorisé
Barème donné à titre indicatif

M. Shapiro, J. Sopena, G. Thomas

Vous rédigerez vos exercices sur deux copies différentes : la première pour les exercices I à III et la seconde pour l'exercice IV.

I. MEMOIRE TRANSACTIONNELLE (5 POINTS)

I.1. (1 points)

Quelles sont les différences entre une mémoire transactionnelle délayée et une mémoire transactionnelle immédiate. Expliquez dans quelle circonstance l'une est meilleures que l'autre.

I.2. (4 points)

On se place dans le cadre d'une mémoire transactionnelle **immédiate au plus tard**. La mémoire est constituée de trois variables x, y et z valant respectivement 3, 5 et 7. Trois threads exécutent les codes suivants :

Thread 1	Thread 2	Thread 3
1. atomic { 2. int t1 = x ; 3. t1 = t1 + y + z ; 4. y = t1 ; 5. }	6. atomic { 7. int t2 = y ; 8. t2 = x + t2 + z ; 9. z = t2 ; 10. }	11. atomic { 12. int t3 = z ; 13. t3 = x + y + t3 ; 14. x = t3 ; 15. }

Un unique processeur exécute séquentiellement les lignes 1, 2, 3, 4, 6, 7, 11, 12, 13, 14, 8, 9, 5, 10, 15. Donnez l'état de la mémoire, des variables t1, t2, t3 et des copies locales des variables à chaque étape. Justifiez et expliquez vos choix lorsque les lignes 5, 10 et 15 sont exécutées.

II. MONITEUR DE MACHINE VIRTUELLE (5 POINTS)

II.1. (1,5 points)

Expliquez ce qu'est la table des page ombre : à quoi elle sert et comment on la met en œuvre.

II.2. (3,5 points)

On suppose qu'on utilise un moniteur de type II. Le système hôte est un Linux et sa table des pages est à l'adresse virtuelle hôte 0xc200'000. Le système invité est un Linux et sa table des pages se trouve à l'adresse virtuelle invitée 0xc200'000.

Expliquez ce qui se passe lorsque le système invité modifie l'adresse 0xc200'0004 et modifie donc sa table des pages ? Vous expliquerez aussi comment la collision d'adresse est évitée.

III. RAMASSE MIETTES (4 POINTS)

III.1. (1 points)

Pour quelle raison ne peut-on pas construire un ramasse-miette conservatif et copiant ?

III.2. (3 points)

Dans l'algorithme générationnel copiant avec un allocateur par thread, est-il possible qu'un objet alloué dans le paradis du thread A ne soit référencé que par le paradis du thread B ? Justifier. Que pouvez vous en conclure sur la collection de la jeune génération ?

IV. STRUCTURES DE DONNEES SANS VERROU (6 POINTS)

Vous devez choisir deux questions parmi les trois proposées.

IV.1. (3 points)

Algorithmes de liste chaînée à grain fin : Dans l'algorithme "paresseux" pourquoi la procédure de validation doit-elle vérifier que `pred.next == current` ? Donner un contre-exemple et justifier logiquement.

IV.2. (3 points)

Algorithmes de liste chaînée à grain fin : l'algorithme "paresseux" et l'algorithme "sans verrou" traitent les nœuds marqués "ignored" de façon différente. Dans le premier, c'est au fil qui marque un nœud de retirer celui-ci de la liste. Dans le second, tout fil qui rencontre un nœud marqué le retire de la liste. Expliquer pourquoi : donner un contre-exemple et justifier logiquement.

IV.3. (3 points)

Algorithme de pile à élimination : montrer qu'un push et un pop éliminés au passage dans le tableau sont bien linéarisables par rapport à tous les autres push et pop.