Université Constantine 2 — Année Universitaire 2022-2023
Faculté des NTIC — Master 1 – Option : RSD
Département IFA — Matière : Algorithmes Distribués

Durée : 1h30 — **Contrôle de connaissances** — **Date : 04/06/2023**

Nom & Prénom : **Groupe :**

QCM (10 pts) : Cochez les bonnes réponses (Tout ou rien).

1) Un algorithme distribué est :
   - ☐ Dit asymétrique si le même code est implémenté sur chaque site du système distribué.
   - ☐ Dit acceptable s'il vérifie toutes les propriétés fonctionnelles
   - ☐ Correcte s'il vérifie les bonnes propriétés de performance
   - ☐ Implémentable s'il est correcte et le système de communication vérifie les hypothèses qu'il requière
2) Soit la solution à base du jeton du problème de la section critique distribuée d'un système dont la topologie est en anneau :
   - ☐ Cette solution est facile à élaborer si les liaisons sont régulières
   - ☐ Cette solution est facile à élaborer si les liaisons sont fiables
   - ☐ La complexité de la solution est due à la perte du jeton malgré que les liaisons soient fiables
   - ☐ Cette solution est sensible au nombre de sites appartenant au système
3) Pour un algorithme distribué d'exclusion mutuelle basé sur les permissions individuelles :
   - ☐ Un site donne sa permission à un autre site en prenant en considération son état uniquement
   - ☐ La réception de toutes les permissions demandées est une condition nécessaire d'accès à la section critique
   - ☐ Le nombre de permissions nécessaires pour l'accès à la section critique est toujours égal à n-1, n est le nombre de sites
   - ☐ La réception de toutes les permissions demandées est toujours inférieure à n.
4) Pour un algorithme distribué d'exclusion mutuelle basé sur les permissions d'arbitre :
   - ☐ Un site donne sa permission à un autre site en prenant en considération les requêtes qu'il a reçu
   - ☐ La réception de toutes les permissions demandées est une condition suffisante d'accès à la section critique
   - ☐ La responsabilité de la gestion de l'accès à la section critique est dite égale pour tous les sites si Card(Ri) = K constante.
   - ☐ L'algorithme est à contrôle centralisé si chaque site est responsable de l'accès à la section critique d'un seul autre site
5) Pour l'algorithme de Carvalho et Roucairol :
   - ☐ Les ensembles Ri sont initialement construits de manière aléatoire
   - ☐ Les ensembles Ri évoluent selon l'algorithme
   - ☐ Les ensembles Ri évoluent selon l'application distribuée qui l'utilise
   - ☐ Les ensembles Ri peuvent devenir tous vides à un moment donné
6) Pour l'algorithme de Misra et Chandy :
   - ☐ La distribution initiale des permissions sur les sites est correcte si le graphe de priorité est cyclique
   - ☐ Les horloges logiques ne sont pas bornées
   - ☐ L'algorithme est adaptatif car les ensembles Ri évoluent selon l'application distribuée qui l'utilise
   - ☐ Un ensemble Ri peut devenir vide à un moment donné
7) Pour un protocole de mise en œuvre des rendez-vous distribués :
   - ☐ Il permet d'implémenter un langage de haut niveau asynchrone de manière synchrone
   - ☐ La propriété de coordination synchrone traduit le principe du tout ou rien
   - ☐ La propriété d'exclusion mutuelle assure l'exécution d'un plus grand nombre de rendez-vous conflictuels
   - ☐ La propriété de vivacité permet que tout rendez-vous possible à un instant donné soit engagé au bout d'un temps fini

8) Le calcul de l'état global passe par le calcul d'une tranche :
   - La tranche permet de donner l'état de tous les processus à un même instant
   - La tranche permet de donner l'état de tous les processus à la l'instant de la fin du calcul
   - Un évènement de la tranche n'est jamais causé par un évènement qui ne lui appartient pas
   - Pour chaque canal (unidirectionnel) c, L'état de c est la séquence des messages émis par les événements appartenant à la tranche, et reçus par les événements qui ne lui appartiennent pas
9) Pour un algorithme de détection de la terminaison distribuée :
   - La propriété de sûreté signifie que si l'application se termine, la terminaison sera sûrement détectée
   - La propriété de sûreté signifie que si l'algorithme détecte la terminaison de l'application, cette dernière s'est réellement terminée
   - La propriété de vivacité signifie que l'algorithme orientera l'évolution de l'application pour qu'elle se termine au bout d'un temps fini
   - La propriété de vivacité signifie que l'algorithme déclarera la terminaison de l'application après un temps fini de son lancement
10) Dans un système utilisant le temps virtuel global :
   - Dans l'approche pessimiste, tout message reçu est systématiquement délivré
   - Le problème de l'interblocage induit par le respect de la contrainte de cohérence est résolu par l'introduction des messages null()
   - Dans l'approche optimiste, un message reçu n'est délivré que si sa livraison respecte la contrainte de cohérence
   - Dans l'approche optimiste, en cas de violation de la contrainte de cohérence, un état global déjà calculé est utilisé pour restituer un état cohérent.

Exercice (10 pts) : Soit un système de **n** sites, $\mathbf{S_0, S_1 \dots S_{n-1}}$, interconnectés par un anneau unidirectionnel dont les liaisons sont fiables. Nous voulons écrire un algorithme distribué qui génère le graphe des marquages d'un réseau de Petri ***R***. Pour simplifier l'écriture de l'algorithme, nous supposons une fonction **Succ** définie par : ***Succ(R,M) = {Mi : Mi est un marquage accessible à partir de M par le franchissement d'une transition de R}***. On utilise la fonction **MD5** tel que pour un marquage **M**, **MD5(M)** donne une valeur numérique entière de cryptage de **M**. La particularité de cette fonction est que deux marquages qui sont syntaxiquement très proches auront des valeurs très dispersées. Un marquage **M** sera stocké sur la machine $\mathbf{S_i}$ tel que $i = MD5(M) \bmod n$.

Principe de l'algorithme : Initialement Un site quelconque est choisi comme initiateur du calcul.

Le site initiateur calcul l'adresse du site qui doit stocker le marquage initial $\mathbf{M_0}$ du réseau de Petri **R**.

Un site qui reçoit un message contenant un marquage à stocker, vérifie si ce dernier est déjà stocké. Dans le cas contraire il le stocke localement et calcul ses états successeurs et fait le traitement nécessaire pour chercher les sites qui devraient les accueillir.

Pour un marquage stocké localement, les transitions qui le lient avec ses marquages successeurs sont créées. Ces transitions peuvent être locales ou distantes. Une transition distante est définie par le nom du site qui héberge son marquage cible (d'arrivée).

1) En suivant la démarche de conception des algorithmes distribués vue en cours, proposer un algorithme distribué qui réalise la génération du graphe des marquages d'un réseau de Petri borné.
2) Proposer une méthode pour statuer sur la terminaison de l'opération de génération du graphe des marquages. La terminaison sera-t-elle détectée au bout d'un temps fini ?
3) Quel est l'intérêt de l'utilisation de la fonction **MD5** ?

Corrigé type

QCM (10 pts) : Cochez les bonnes réponses (Tout ou rien).

1) Un algorithme distribué est :
   - ☐ Dit asymétrique si le même code est implémenté sur chaque site du système distribué.
   - ☐ Dit acceptable s'il vérifie toutes les propriétés fonctionnelles
   - ☐ Correcte s'il vérifie les bonnes propriétés de performance
   - ✓ Implémentable s'il est correcte et le système de communication vérifie les hypothèses qu'il requière
2) Soit la solution à base du jeton du problème de la section critique distribuée d'un système dont la topologie est en anneau :
   - ☐ Cette solution est facile à élaborer si les liaisons sont régulières
   - ✓ Cette solution est facile à élaborer si les liaisons sont fiables
   - ☐ La complexité de la solution est due à la perte du jeton malgré que les liaisons soient fiables
   - ✓ Cette solution est sensible au nombre de sites appartenant au système
3) Pour un algorithme distribué d'exclusion mutuelle basé sur les permissions individuelles :
   - ✓ Un site donne sa permission à un autre site en prenant en considération son état uniquement
   - ✓ La réception de toutes les permissions demandées est une condition nécessaire d'accès à la section critique
   - ✓ Le nombre de permissions nécessaires pour l'accès à la section critique est toujours égal à n-1, n est le nombre de sites
   - ✓ La réception de toutes les permissions demandées est toujours inférieure à n.
4) Pour un algorithme distribué d'exclusion mutuelle basé sur les permissions d'arbitre :
   - ✓ Un site donne sa permission à un autre site en prenant en considération les requêtes qu'il a reçu
   - ☐ La réception de toutes les permissions demandées est une condition suffisante d'accès à la section critique
   - ☐ La responsabilité de la gestion de l'accès à la section critique est dite égale pour tous les sites si Card(Ri) = K constante.
   - ☐ L'algorithme est à contrôle centralisé si chaque site est responsable de l'accès à la section critique d'un seul autre site
5) Pour l'algorithme de Carvalho et Roucairol :
   - ✓ Les ensembles Ri sont initialement construits de manière aléatoire
   - ☐ Les ensembles Ri évoluent selon l'algorithme
   - ✓ Les ensembles Ri évoluent selon l'application distribuée qui l'utilise
   - ☐ Les ensembles Ri peuvent devenir tous vides à un moment donné
6) Pour l'algorithme de Misra et Chandy :
   - ☐ La distribution initiale des permissions sur les sites est correcte si le graphe de priorité est cyclique
   - ☐ Les horloges logiques ne sont pas bornées
   - ✓ L'algorithme est adaptatif car les ensembles Ri évoluent selon l'application distribuée qui l'utilise
   - ✓ Un ensemble Ri peut devenir vide à un moment donné
7) Pour un protocole de mise en œuvre des rendez-vous distribués :
   - ☐ Il permet d'implémenter un langage de haut niveau asynchrone de manière synchrone
   - ✓ La propriété de coordination synchrone traduit le principe du tout ou rien
   - ☐ La propriété d'exclusion mutuelle assure l'exécution d'un plus grand nombre de rendez-vous conflictuels
   - ☐ La propriété de vivacité permet que tout rendez-vous possible à un instant donné soit engagé au bout d'un temps fini

8) Le calcul de l'état global passe par le calcul d'une tranche :
   - ☐ La tranche permet de donner l'état de tous les processus à un même instant
   - ☐ La tranche permet de donner l'état de tous les processus à la l'instant de la fin du calcul
   - ✓ Un évènement de la tranche n'est jamais causé par un évènement qui ne lui appartient pas
   - ✓ Pour chaque canal (unidirectionnel) c, L'état de c est la séquence des messages émis par les événements appartenant à la tranche, et reçus par les événements qui ne lui appartiennent pas

9) Pour un algorithme de détection de la terminaison distribuée :
   - ☐ La propriété de sûreté signifie que si l'application se termine, la terminaison sera sûrement détectée
   - ✓ La propriété de sûreté signifie que si l'algorithme détecte la terminaison de l'application, cette dernière s'est réellement terminée
   - ☐ La propriété de vivacité signifie que l'algorithme orientera l'évolution de l'application pour qu'elle se termine au bout d'un temps fini
   - ☐ La propriété de vivacité signifie que l'algorithme déclarera la terminaison de l'application après un temps fini de son lancement

10) Dans un système utilisant le temps virtuel global :
   - ☐ Dans l'approche pessimiste, tout message reçu est systématiquement délivré
   - ✓ Le problème de l'interblocage induit par le respect de la contrainte de cohérence est résolu par l'introduction des messages null()
   - ☐ Dans l'approche optimiste, un message reçu n'est délivré que si sa livraison respecte la contrainte de cohérence
   - ✓ Dans l'approche optimiste, en cas de violation de la contrainte de cohérence, un état global déjà calculé est utilisé pour restituer un état cohérent.

Exercice (10 pts) :

Question1) En suivant la démarche de conception des algorithmes distribués vue en cours, proposer un algorithme distribué qui réalise la génération du graphe des marquages d'un réseau de Petri **borné**.

1. ***Types de processus qui composent le système : (0,5)***

Nous distinguons deux types de processus

- ✓ L'initiateur : Qui a pour tâche de lancer le calcul du graphe des marquages
- ✓ Les processus associés aux sites qui reçoivent les requêtes pour stocker localement un marquage et calculer ses marquages successeurs. L'initiateur fait aussi parti de ces processus.

2. ***Evènements***

- ***Evènements qui sont à l'initiative des processus : (0,5)***
  - Pour l'initiateur : Evènement de lancement du calcul du graphe des marquages
  - Pour les autres processus : Pas d'évènements
- ***Evènements subis par les processus : (0,5)***
  - Tous les processus doivent réagir de la même manière à la réception d'une requête de stockage local d'un marquage et le traitement de ses marquages successeurs

3. ***Procédures associées aux évènements :***

- **Pour l'initiateur**

Lors d'un appel à générer graphe des marquages (R, M_0) (1,5)

```
Début
    Si MD5(M0) mod n = mon identificateur Alors
        Créer nouvel état M0 ; /* M0 est stocké localement
        Traiter(M0) ;
    Sinon
        Envoyer Requête(M0,MD5(M0) mod n) au suivant ;
    FinSi
Fin
```

- **Pour tous les sites**

Les procédures sont écrites pour un site K

Lors de la réception de Requête(M,l) (1,5)

```
Début
    Si not (l=k) Alors
        Envoyer Requête(M,l) au suivant ;
    Sinon
        Si M n'existe pas localement Alors
            Créer nouvel état M ; /* M est stocké localement
            Traiter(M) ;
        FinSi
    FinSi
Fin
```

Procédure Traiter (M : Marquage) (1,5)

```
Début
    Pour Mi ∈ Succ(R,M) faire
        Si MD5(Mi) mod n = mon identificateur Alors
            Créer nouvel état Mi ; /* Mi est stocké localement
            Créer transition (M,t,Mi,K) / M[t>Mi ;
            Traiter (Mi) ;
        Sinon
            Var Dest : entier = MD5(Mi) mod n ;
            Créer transition (M,t,Mi,Dest) / M[t>Mi ; /*Transition inter-sites
            Envoyer Requête (Mi) à Dest ;
        FinSi
```

FinPour

Fin

Le graphe des marquages étant distribué, le successeur d'un marquage peut :

- ✓ Soit stocké sur le même site, dans ce cas la transition est étiquetée par l'identificateur du même site.
- ✓ Soit stocké sur un site distant, dans ce cas la transition est étiquetée par l'identificateur du site qui accueillera le marquage résultant.

4. Propriétés fonctionnelles de l'algorithme

- ***Propriété de sûreté (0,5):*** Pour toute transition **(M_i,t,Mj,K)** du graphe de marquages distribué correspond une transition **(M_i,t,Mj)** du graphe des marquages. En d'autres termes, il n'existe pas de marquages et de transitions imaginaires dans le graphe des marquages distribué généré.
- ***Propriété de vivacité (0,5):*** Pour toute transition **(M_i,t,Mj)** du graphe des marquages, l'algorithme finira par construire une **(M_i,t,Mj,K)** du graphe de marquages distribué tel que M_i est sur le site MD5(M_i) mod n et M_j est sur le site K=MD5(M_j) mod n.

  Preuve (0,5): Trivial car la génération distribuée est basée sur la sémantique du modèle des réseaux de Petri. La seule différence concerne les transitions intersites. Ces dernières préservent le même sens des transitions correspondantes du graphe des marquages centralisé.

Question 2) Proposer une méthode pour statuer sur la terminaison de l'opération de génération du graphe des marquages. La terminaison sera-t-elle détectée au bout d'un temps fini ?

1. Pour statuer sur la terminaison nous pouvons faire appel à un algorithme de détection de la terminaison qui fonctionne sur une topologie en anneau dont les liaisons sont fiables. Deux types d'algorithmes peuvent être utilisés :
   1.1. (0,5) L'algorithme est invoqué par un site à un moment donné lorsque le site soupçonne que l'algorithme à terminé la génération du graphe des marquages. Par exemple, lorsque le site à terminé la génération localement et qu'il n'a pas reçu de requête pendant une certaine durée prédéfinie.
   1.2. (0,5) L'algorithme de détection de la terminaison est tout le temps en exécution. Il termine lorsqu'il détecte la terminaison.
2. (0,5) Etant donné que la génération du graphe des marquages concerne un réseau de Petri borné, le graphe des marquages est de taille finie. De ce fait, la génération se terminera surement au bout d'un temps fini. Il est clair, qu'à partir de l'instant de terminaison de l'algorithme de génération du h=graphe des marquages, l'algorithme de détection de la terminaison détectera cette terminaison car il vérifie la propriété de vivacité.

Question 3) (1) Quel est l'intérêt de l'utilisation de la fonction **MD5** ?

Cette fonction permet d'obtenir un équilibrage de la charge de stockage des fragments du graphe des marquages sur les différents sites. En d'autres termes, les fragments du graphe distribués sont approximativement de même taille.