Université de Batna 2 **Batna le 24 Janvier 2018**
Faculté des Mathématiques et d'Informatique
Département d'Informatique
Section : Master 2 RSD
Durée : 90 minutes

EXAMEN FINAL
MODULE : ALGORITHMES DISTRIBUES

Questions de Cours (06 Points) :

1- Quels sont les inconvénients d'un système distribué asynchrone ?
2- Est-il possible d'utiliser les sémaphores comme un moyen d'exclusion mutuelle dans les systèmes distribués Asynchrones avec échange de messages ?
3- Quelle est la différence entre les deux protocoles de Validation Atomique 2PC et 3PC ?
4- Définir les trois propriétés qui caractérisent, formellement, le problème du consensus.
5- Comment contourner le résultat d'impossibilité de FLP ?
6- Donner les propriétés qui caractérisent un détecteur de défaillances de la classe ◊**S** ?
7- Quelle est la relation entre un détecteur de défaillances de la classe ◊*S* et un autre de la classe ◊W ?

Exercice 02 (07 Points) :

On considère le problème de la k-exclusion mutuelle qu'est une généralisation du problème de l'exclusion mutuelle. Cette généralisation consiste à imposer que, sur les ***N*** processus du système réparti, seuls **k≥1** processus peuvent être en section critique simultanément. (Dans l'exclusion mutuelle normale **k= 1**).

Pour traiter ce problème, nous proposons de modifier l'algorithme de Ricart et Agrawala. La modification proposée est la suivante : on attend toujours ***(N – 1)*** autorisations, mais les processus qui sont en section critique (ou qui attendent d'y rentrer) peuvent distribuer au maximum ***(k– 1)*** autorisations avant de stocker les demandes dans la file d'attente comme dans l'algorithme original.

Chaque processus ***Pi*** qui demande l'entrée en section critique initialise donc un compteur $compteur_i$ à 1. Ce compteur représente le nombre de processus qui ont demandé l'entrée en section critique après lui et qu'il a autorisé à y rentrer (lui compris).

Lorsqu'un processus ***Pj*** demande l'autorisation d'entrer en section critique à un processus ***Pi*** qui est en section critique (ou en attente de section critique), il y a deux possibilités :

1- Soit la demande de ***Pj*** précède celle de ***Pi***, auquel cas ***Pi*** envoie une permission à ***Pj*** sans modifier $compteur_i$.

2- Soit la demande de ***Pi*** précède celle de ***Pj*** :

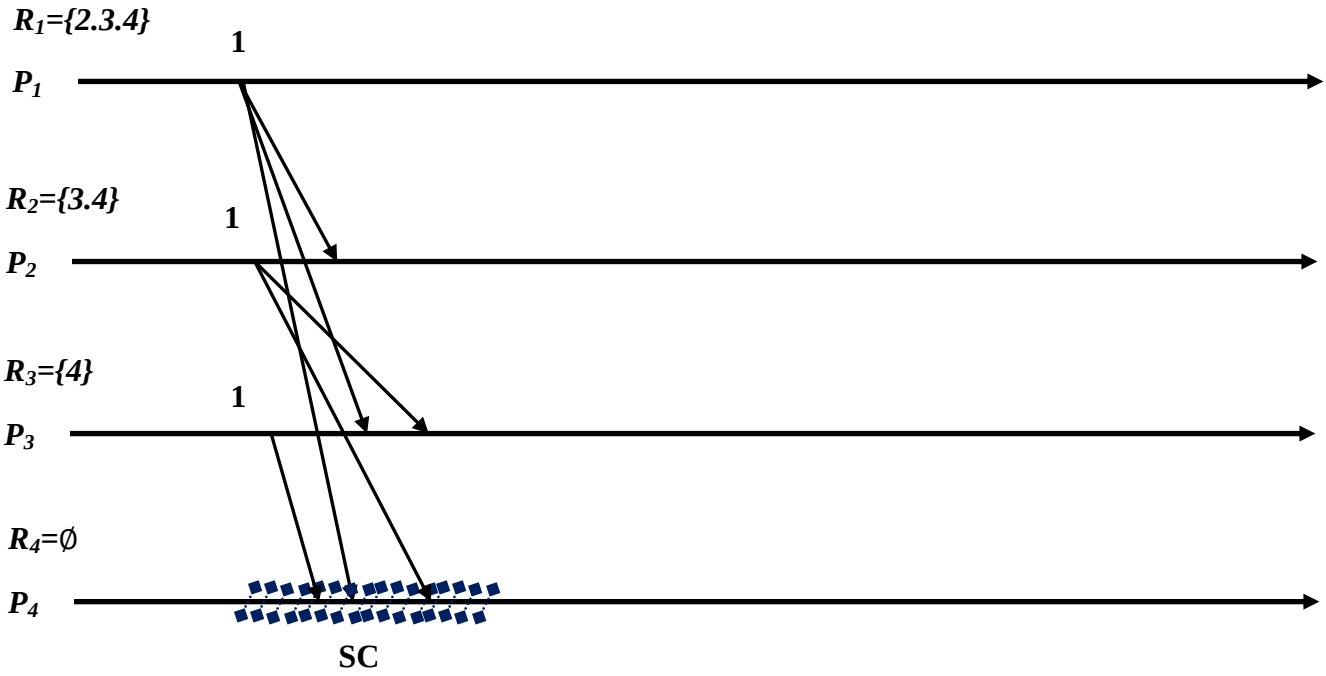
- Si $compteur_i < K$ alors ***Pi*** envoie une permission à ***Pj*** et incrémente $compteur_i$.
- Si $compteur_i = K$ alors ***Pi*** met la demande de ***Pj*** dans sa file d'attente comme dans l'algorithme original.

Q1 : En se basant sur l'algorithme de Ricart et Agrawala, écrire un algorithme qui prend en charge cette modification.

Q2: Est-il possible de résoudre le problème de la k-exclusion mutuelle en modifiant l'algorithme de Carvalho-Roucairol ? Si oui, expliquer le principe à suivre pour le modifier.

Exercice 03 (04 Points) :

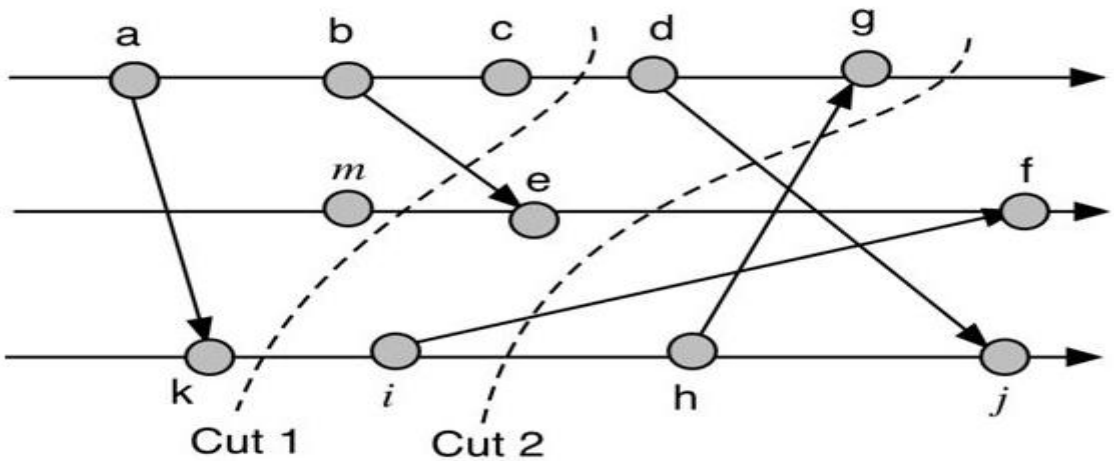
Appliquez algorithme de Carvalho-Roucairol sur le diagramme suivant et déterminez l'ordre d'entrée des processus en section critique.



Exercice 04 (03 Points) :

Q1-Définir la notion de coupure cohérente. Pour l'exécution d'un calcul réparti proposé dans la figure ci-dessous, identifier les coupures cohérentes. Justifier votre réponse.

Q2- Quelles sont les modifications qu'il faut apporter à l'algorithme de Chandy-Lamport pour enregistré un état global fortement cohérent (c'est-à-dire, tous les états de canaux sont enregistrés vide) ?



GOOD LUCK