

Corrigé Type d'examen de SD2

Questions du Cours : (04 points)

Q-1) Quel est le rôle des horloges logiques ou vectorielles dans les systèmes distribués ? (01 points)

R-1) Le rôle des horloges logiques est de définir un ordre partiel ou global entre les événements compatibles avec la relation de précedence causale. Pour cela, on effectue une datation des événements.

Q-2) Quel est le profit d'utiliser un algorithme à base de permission par arbitre ? (01 points)

R-2) Le profit d'utiliser l'algorithme à base de permission par arbitre c'est de minimiser le nombre de messages échangés entre les différents sites.

Q-3) Combien de cycles sont nécessaire afin de sélectionner un arbitre dans l'algorithme d'élection de Chang et Roberts dans un anneau logique unidirectionnel ? (01 points)

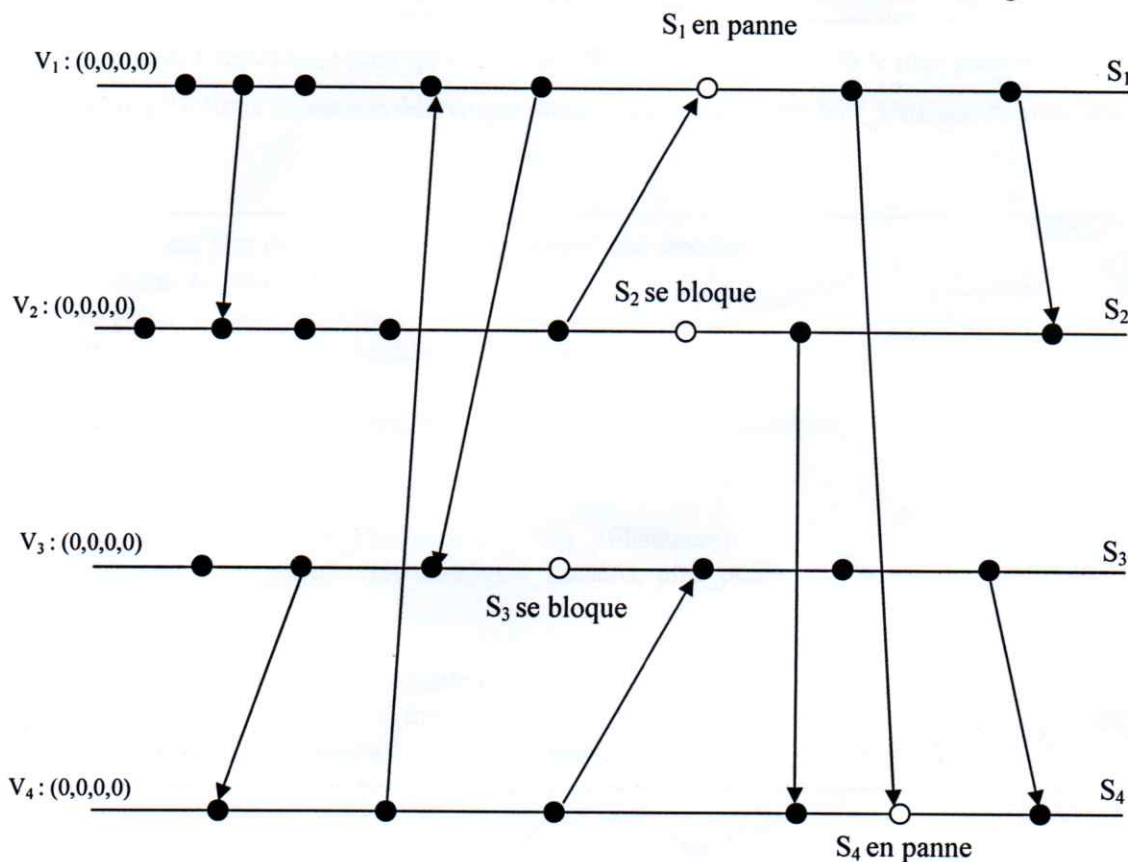
R-3) Le nombre de cycles nécessaire afin de sélectionner un arbitre dans l'algorithme de Chang et Roberts dans un anneau logique unidirectionnel est : 3.

Q-4) Combien de cycles pour informer les autres sites du résultat d'élection ? (01 points)

R-4) le nombre de cycles pour informer les sites est 1.

Exercice N° 1 : (10 points)

1) Soit le schéma ci-dessous présentant l'interaction entre 04 sites utilisant des horloges vectorielles :



Q-1) Compléter le schéma par les valeurs des horloges et estampillages correcte ?

➤ **Remarque** : les valeurs des horloges gardent leurs valeurs en cas de panne ou blocage.

Exercice N° 2 : (06 Points)

Soit l'algorithme d'élection suivant :

texte exécuté lors de la décision de provoquer une élection

```
participant ← vrai ;
plus_grand ← mon_numéro ;
envoyer_voisin_gauche (élection, mon_numéro) ;
```

texte exécuté lors de la réception de (élection, j)

```
si non participant,
    participant ← vrai ;
    plus_grand ← max(mon_numéro, j).
    envoyer_voisin_gauche (élection, plus_grand) ;
sinon
    si j > plus_grand,
        plus_grand ← j ;
        envoyer_voisin_gauche (élection, plus_grand) ;
    sinon si j = mon_numéro, envoyer_voisin_gauche (élu, mon_numéro);
```

texte exécuté lors de la réception de (élu, j)

```
vainqueur ← j ;
participant ← faux ; (* initialisation pour la prochaine élection *)
si j ≠ mon_numéro, envoyer_voisin_gauche (élu, j) ;
```

Q-1) Modifier l'algorithme pour qu'il fait l'élection du site qui possède le plus petit nombre d'utilisateur ? (Nous allons considérer que nous avons une variable Nbr_Utilisateurs pour chaque site)

R-1)

texte exécuté lors de la décision de provoquer une élection

```
participant ← vrai ;
plus_grand ← mon_numéro ;
envoyer_voisin_gauche (élection, mon_numéro, Nbr_Utilisateurs) ;
```

texte exécuté lors de la réception de (élection, j, j_Nbr_utilisateurs)

```
si non participant,
    participant ← vrai ;
    plus_petit ← min(Nbr_Utilisateurs, j_Nbr_utilisateurs).
    envoyer_voisin_gauche (élection, mon_numéro, plus_petit) ;
sinon
    si j < plus_petit,
        plus_petit ← j_Nbr_utilisateurs;
        envoyer_voisin_gauche (élection, mon_numéro, plus_petit) ;
    sinon si j = mon_numéro, envoyer_voisin_gauche (élu, mon_numéro);
```

texte exécuté lors de la réception de (élu, j)

vainqueur ← j ;

participant ← faux ; (* initialisation pour la prochaine élection *)

si $j \neq \text{mon_numéro}$, envoyer_voisin_gauche (élu, j) ;