

Examen de SD2

Questions du Cours : (06 points)

Q-1) Pourquoi on utilise les horloges logiques dans les SDs?

{لماذا نستخدم الساعات المنطقية في الأنظمة الموزعة}

R-1) On utilise les horloges logiques dans les SDs afin de définir un ordre partiel ou total entre les évènements des différents sites. (02 points)

Q-2) Combien de cycles sont nécessaire pour faire l'élection ?

{كم دورة تلزم لتطبيق خوارزمية الانتخاب}

R-2) Le nombre de cycles nécessaire pour faire une élection est : 03 (02 points)

Q-3) Pourquoi on calcul un état global du SD ?

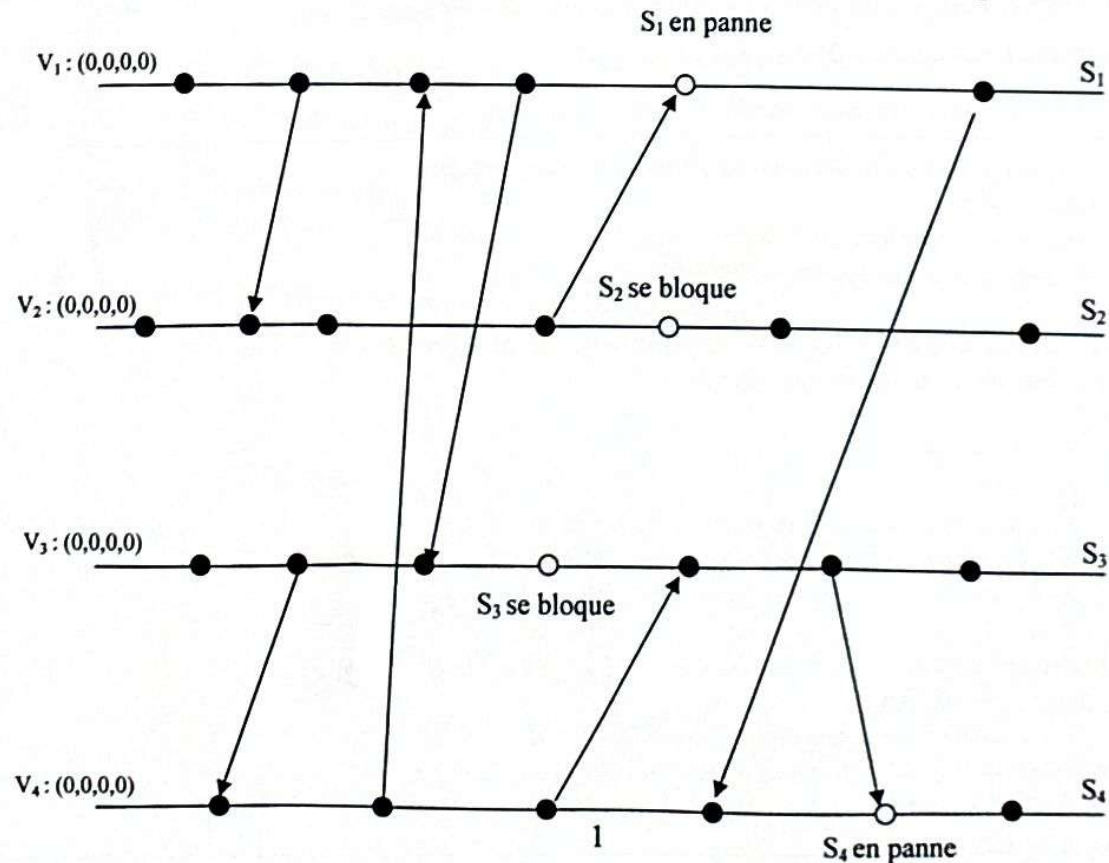
{لماذا نقوم بحساب الحالة العامة للنظام الموزع}

R-3) On calcul l'état global d'un SD afin de:

- Définir un point de reprise du SD, ou
- Analyser le SD.

Exercice N° 1 : (08 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 sont réinitialisées en cas de panne ou blocage.

{القيم ترجع إلى الصفر في حالة عطب أو توقف للنظام}

Exercice N° 2 : (06 Points)

Soit l'algorithme de parcours suivant :

texte exécuté lors de la décision de Pr de lancer un parcours

marqué ← vrai ;

nb_attendus ← cardinal (voisins) ;

∀ j ∈ voisins, envoyer (parcours, info, r) à Pj ;

texte exécuté lors de la réception de (parcours, info, j)

si marqué, nb_attendus ← nb_attendus - 1 ;

sinon

marqué ← vrai ;

père ← j ;

nb_attendus ← cardinal (voisins) - 1 ;

∀ j ∈ voisins - père, envoyer (parcours, info, i) à Pj ;

si nb_attendus = 0, envoyer (retour, info') à père

texte exécuté lors de la réception de (retour, info') par Pi

nb_attendus ← nb_attendus - 1 ;

actualiser info' ;

si nb_attendus = 0, si i T r, envoyer (retour, info') à père;

sinon terminé ← vrai ;

Q-1) Modifier l'algorithme pour qu'il calcul le nombre des sites actif ?

{قم بتغيير الخوارزمية بحيث تقوم بحساب عدد المواقع النشطة}

R-1) Soit le variable Nbr_Sites_Actif=0 (pour chaque site);

texte exécuté lors de la décision de Pr de lancer un parcours

marqué ← vrai ;

nb_attendus ← cardinal (voisins) ;

∀ j ∈ voisins, envoyer (parcours, Nbr_Sites_Actif, r) à Pj ;

texte exécuté lors de la réception de (parcours, Nbr_Sites_Actif_J, j)

si marqué, nb_attendus ← nb_attendus - 1 ;

sinon

marqué ← vrai ;

père ← j ;

nb_attendus ← cardinal (voisins) - 1 ;

∀ j ∈ voisins - père, envoyer (parcours, Nbr_Sites_Actif, i) à Pj ;

si nb_attendus = 0, envoyer (retour, Nbr_Sites_Actif) à père

texte exécuté lors de la réception de (retour, Nbr_Sites_Actif_J) par Pi

nb_attendus ← nb_attendus - 1 ;

Nbr_Sites_Actif = Nbr_Sites_Actif + Nbr_Sites_Actif_J;

si nb_attendus = 0, si i T r, envoyer (retour, Nbr_Sites_Actif) à père;

sinon terminé ← vrai ;