

Examen semestriel

Module de Systèmes d'exploitation

Corrigé

Exercice 1 : (7 points)

On considère le problème classique de synchronisation des "Lecteurs-Rédacteurs", avec priorité aux Lecteurs, mais en ajoutant la règle suivante : les Lecteurs présents dans le fichier peuvent être plusieurs mais ne doivent pas dépasser N (un certain entier).

Donnez la solution à ce problème en utilisant les sémaphores.

Réponse :

Wrt : semaphore (init=1), ReadCount : entier (init=0), Mutex : semaphore (init=1), SMax : semaphore (init à NMax),

<pre>Processus Lecteur Début Wait(Mutex) ; ReadCount :=ReadCount+1 ; Si ReadCount=N Alors Wait(Wrt) Finsi ; Signal(Mutex) ; Wait (SMax) /* Lire dans le fichier Signal(SMax); Wait(Mutex) ; ReadCount :=ReadCount-1 ; Si ReadCount=0 Alors Signal(Wrt) Finsi Signal(Mutex) Fin.</pre>	<pre>Processus Redacteur Début Wait(Wrt) /* Ecrire dans le fichier Signal(Wrt) ; Fin.</pre>
---	--

(7 points)

Exercice 2 : (7 points)

Reprendre le même problème , mais en utilisant les moniteurs de Hoare :

Réponse :

M : étant une instance du moniteur.

<pre>lecteur_rédacteur : moniteur ; . var ecr : booléen ; . nl : entier ; . I : entier ; . c_ecr, c_lect : condition ; NMax : entier; C_LectMax : condition; procédure début_lire ; début nl := nl + 1; si ecr alors . c_lect.attendre ; . c_lect.signaler ; I:=I+1 Si I=NMax alors c_lectMax.wait fin ; procédure fin_lire ; début I:=I - 1; c_lectMax.signal nl := nl - 1; si nl = 0 alors . c_ecr.signaler ; fin ; procédure début_écrire ; début si ecr ou nl > 0 alors . c_ecr.attendre ; ecr := vrai ; fin ; procédure fin_écrire ; début ecr := faux ; si nl > 0 alors . c_lect.signaler ; sinon . c_ecr.signaler ; fin ; début ecr := faux ; nl := 0 ; fin</pre>	<pre>Processus Rédacteur Début M. Debut_écrire /* Ecrire dans le fichier */ M.Fin_écrire Fin Processus Lecteur Début M. Debut_Lire /* Lire dans le fichier */ M.Fin_Lire Fin.</pre>
--	--

(7 points)

Exercice 3 : (6 points)

Question 1 : Souvent, les processus sont obligés de communiquer entre eux . Dites quelles sont les principales situations :

Réponse :

Coopération : Les processus coopèrent pour la réalisation d'une certaine tâche. Dans ce cas ils doivent communiquer pour échanger des données.

Concurrence : Les processus sont en compétition pour l'occupation d'une certaine ressource. Les processus doivent communiquer pour régler l'accès à cette ressource.

(1 points)

Question 2 : Expliquez le principe de la méthode de communication dite "mémoire partagée".

Réponse :

Les processus voulant communiquer réservent en mémoire une zone (mémoire partagée) qui servira de "boîte aux lettres" . Un processus qui veut envoyer un message à un autre va le déposer dans la boîte aux lettres. Le processus destinataire ira le récupérer dans la boîte.

(1 points)

Question 3 : Dans la méthode de communication de la "Mémoire partagée", que doit contenir un message au minimum ?

Réponse :

Le message doit contenir 3 éléments au minimum : l'identifiant du processus émetteur, l'identifiant du processus destinataire et le message lui-même.

(1 points)

Question 4 : Dans la méthode de communication de la "Mémoire partagée", lorsqu'un processus P_i envoie un message à un autre processus P_j , dites comment ce dernier peut-il savoir qu'il y'a un message qui lui a été envoyé.

Réponse :

Le processus destinataire va consulter régulièrement le contenu de la boîte aux lettres.

(1 points)

Question 5 : Dans la méthode de communication de la "Mémoire partagée", expliquez comment un processus peut "diffuser" un message. Expliquez le problème qui se pose alors, et donnez une solution.

Réponse :

Le message diffusé comporte dans sa partie destinataire une marque qui symbolise le fait que le message concerne tous les processus (par exemple une étoile *).

Le problème est que le message déposé dans la boîte doit être lu par tous les processus qui iront le lire à des temps différents. Comment alors savoir que le message a été lu par tous les processus ?

La solution serait d'utiliser un compteur de processus ayant lu le message. Lorsque ce compteur est égal à N (le nombre de processus) , on peut alors savoir que le message a été lu par tous les processus et donc il peut être supprimé de la boîte.

(2 points)