Université de Chlef
Département Informatique
Filière : 1 ère Année Master

Mars 2014

Examen semestriel

Module de Systèmes d'exploitation

Corrigé

Exercice 1 :

Pour gérer les ressources d'un système, on adopte le protocole suivant : On numérote tous les types de ressources selon un ordre croissant, par exemple :

1 : Lecteur de disquette
2 : Lecteur USB
3 : Processeur
4 : Imprimante,
…etc.

Et on impose à chaque processus qui demande des ressources de les demander dans l'ordre croissant de leurs numéros. Par exemple, si un processus désire utiliser un processeur, un lecteur USB et une imprimante; il doit les demander dans l'ordre : lecteur-USB (n° 2), Processeur (n°3) et imprimante (n°4).

Question 1 : Démontrez qu'on ne peut pas avoir d'interblocage avec cette méthode.

Réponse :

On ne peut pas avoir d'interblocage avec ce protocole, car l'une des conditions nécessaires et suffisantes pour l'apparition d'interblocage serait non vérifiée : la condition de "l'attente circulaire".

Justification : On fait un raisonnement par l'absurde.

Supposons qu'il y'a un interblocage entre n Processus P0, P1, … Pn-1, qui sont liés par une chaine d'attente circulaire. Le processus P0 possédant une ressource R0 est en attente d'au moins une ressource R1 de P1 (avec n° R0 > n° R1 pour respecter la méthode), le processus P1 est au moins en attente d'une ressource R2 de P2 (avec n° R1> n° R2) , et ainsi de suite … jusqu'au processus Pn-1 possédant une ressource Rn-1 et attendant une ressource R0 détenue par le processus P0 (avec n° R0 > n° Rn-1). D'où une contradiction. L'hypothèse de départ étant fausse, on n'a pas donc d'interblocage

(6 points)

Question 2 : Quel est l'inconvénient de cette méthode ?

Réponse :

Cette méthode peut conduire à une immobilisation inutile des ressources. Reprenons l'exemple de l'énoncé (bande magnétique =1, processeur =2, imprimante=3, …). Supposons qu'un processus P a besoin d'abord du processeur pendant X unité de temps, ensuite du lecteur de bande magnétique pendant Y unités de temps et enfin de l'imprimante . Pour respecter la méthode, P doit d'abord demander le lecteur de bande magnétique qu'il immobilisera inutilement pendant X temps, et le processeur. Après l'utilisation effective du processeur , du lecteur de bande magnétique et de l'imprimante, le processeur termine.

(4 points)

Exercice 2 :

Un stade d'athlétisme peut recevoir les athlètes de trois (3) clubs A, B et C qui viennent s'y entraîner. Pour organiser les entraînements, on impose la règle suivante :

A un instant donné, le stade peut recevoir un nombre quelconque d'athlètes mais de deux clubs au maximum. Par exemple, 5 athlètes du club B et 3 athlètes du Club C peuvent s'entraîner en même temps, mais si un athlète du club A veut accéder au stade, il doit attendre jusqu'à ce que tous les athlètes aient quitté le stade, soit du club B soit du club C.

On vous demande de proposer un schéma de synchronisation des processus : Processus A, Processus B et Processus C correspondant respectivement à des athlètes des clubs A, B et C, et ce en utilisant des sémaphores.

Déclarez clairement vos variables et précisez leurs initialisations.

Réponse :

Déclarations :
NbA, nbB, nbC : entier (initialisés à 0) représentant respectivement le nombre d'athlètes des équipes A, B et C.
mutex_A, murex_B, mutex_C : sémaphore (initialisés à 1) permettant l'accès en mutex aux variables partagées nbA, nbB et nbC.
OK : semaphore (initialisé à 2) assure la présence d'athlètes d'au plus 2 équipes.

Processus A	Processus B	Processus C
Début	Début	Début
Wait(mutex_A)	Wait(mutex_B)	Wait(mutex_C)
Si (nbA=0) Alors wait(OK)	Si (nbB=0) Alors wait(OK)	Si (nbC=0) Alors wait(OK)
Finsi	Finsi	Finsi
NbA++;	NbB++;	NbC++;
Signal(mutex_A)	Signal(mutex_B)	Signal(mutex_C)
//entrer au stade et s'entraîner	//entrer au stade et s'entraîner	//entrer au stade et s'entraîner
wait(mutex_A)	wait(mutex_B)	wait(mutex_C)
nbA--;	nbB--;	nbC--;
si (nbA=0) alors signal(OK)	si (nbB=0) alors signal(OK)	si (nbC=0) alors signal(OK)
finsi	finsi	finsi
signal(mutex_A)	signal(mutex_B)	signal(mutex_C)
fin.	fin.	fin.

(10 points)