

Examen semestriel

Module de « Systèmes d'exploitation2 »

Corrigé

Exercice 1 :

Question 1 : Expliquez le principe de la communication par « mémoire partagée ».

Réponse :

La mémoire partagée permet à deux processus ou plus d'accéder à la même zone mémoire comme s'ils avaient leurs pointeurs dirigés vers le même espace mémoire. Lorsqu'un processus modifie la mémoire, tous les autres processus voient la modification.

Pour utiliser un segment de mémoire partagée, un processus doit allouer le segment. Puis, chaque processus désirant accéder au segment doit l'attacher. Après avoir fini d'utiliser le segment, chaque processus le détache. À un moment ou à un autre, un processus doit libérer le segment.

(4 points)

Question 2 : Expliquez la différence entre la communication par « mémoire partagée » et la « mémoire mappée ».

Réponse :

La « mémoire mappée » est un mode de communication analogue à la « mémoire partagée », sauf que la communication se fait par un fichier et non par zone partagée.

(3 points)

Exercice 2 : On reprend le problème du coiffeur endormi vu en TD, mais en supposant que le nombre de places du salon est illimité. . Réécrire la solution de ce problème.

Réponse :

<i>Processus Coiffeur</i>	<i>Processus Client</i>
Début	Début
Cycle	Signal(SClient)
Wait(SClient)	Wait(SCoiffeur)
Signal(SCoiffeur)	Obtenir_coupe
Couper_cheveux	Fin
FinCycle	
Fin	

Déclarations :

SClient, SCoiffeur :
sémaphores initialisés à
0.

(5 points)

Exercice 3 : Proposez une solution au problème des lecteurs-rédacteurs en donnant une priorité aux rédacteurs en utilisant les moniteurs de Hoare.

Réponse :

<pre> Moniteur = Fichier Var Ecrire, Lire : Boolean ; NbLecteur , NbRedacteur : Integer ; CEcrire, CLire : Condition ; Procedure Entry DebutLire Begin If Ecrire Or NBRedacteur>0 then CLire.wait; NbLecteur := NbLecteur +1 ; Lire := True ; CLire.signal End ; Procedure Entry FinLire Begin NbLecteur := NBlecteur – 1 ; If NBlecteur = 0 then Begin Lire :=False ; CEcrire.Signal End End Procedure Entry DebutEcrire Begin NBRedacteur := NBRedacteur + 1 ; If (Ecrire) or (Lire) Then CEcrire.wait Ecrire := True End Procedure Entry FinEcrire Begin Ecrire :=False NBRedacteur := NBRedacteur - 1 If NBRedacteur>0 then CEcrire.Signal Else CLire.Signal End Begin /* Initialisation */ Ecrire :=False ; Lire := False ; NbLecteur := 0 ; NBRedacteur :=0 End. </pre>	<pre> Processus Lecteur Debut Fichier.DebutLire Lire Fichier.FinLire Fin. Processus Redacteur Debut Fichier.DebutEcrire Ecrire Fichier.FinEcrire Fin. </pre>
---	---

(08 Points)