

EXERCICES DIRIGES

Synchronisation de processus

Exercice 1 – Exclusion mutuelle

On considère que lorsqu'un processus réalise une opération d'écriture avec le disque, il l'effectue en appelant une fonction du système ECRIRE_DISQUE (tampon, nb_octets), qui écrit les nb_octets de données placées dans le tableau tampon. Cette fonction système fait appel à une fonction bas niveau du pilote disque Ecrire_blocdisque qui effectue l'écriture physique par bloc de 512 octets.

Le code simplifié de la fonction du système ECRIRE_DISQUE est le suivant :

```
Fonction ECRIRE_DISQUE (tampon, nb_octets)
char tampon[] ;
int nb_octets
{
i = 0 ;
Tant que (nb_octets > 0)
Faire
-- l'écriture physique s'effectue par blocs de 512 octets
    Ecrire_blocdisque (tampon[i, i + 512], 512) ; -- on écrit
physiquement 512 octets
    nb_octets = nb_octets - 512 ; -- on décrémente le nombre total
d'octets à écrire
    i = i + 512 ; -- on avance l'index dans le tableau des octets à
écrire
Fait
Return ;
}
```

L'écriture complète des nb_octets de données placées dans le tableau tampon, réalisée par la fonction système ECRIRE_DISQUE constitue une section critique.

Question 1

Qu'est-ce que cela veut dire ?

Question 2

On dispose de deux primitives de synchronisation suivantes agissant sur un objet Verrou :

- Objet Verrou (état libre, état occupé) ; Verrou est initialement dans l'état libre.
- Verrouiller (Verrou) : si Verrou est dans l'état libre alors le mettre à l'état occupé. Si Verrou est dans l'état occupé, alors bloquer le processus effectuant l'opération Verrouiller.
- Deverrouiller (Verrou) : si un processus est bloqué sur Verrou, le réveiller. Sinon, mettre Verrou à l'état libre.

Utilisez ces deux primitives et l'objet Verrou pour apporter la propriété d'exclusion mutuelle au code de la fonction système ECRIRE_DISQUE.

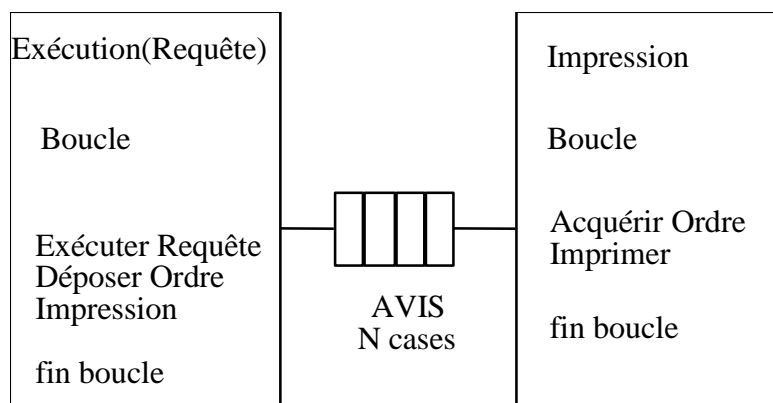
Question 3

Etant donné que la fonction ECRIRE_DISQUE est une fonction du système, celle-ci s'exécute donc en mode superviseur. Quelles opérations aurait-on pu utiliser dans ce cadre pour apporter la propriété d'exclusion mutuelle à la fonction ECRIRE_DISQUE ?

Exercice 2 : Producteur / Consommateur

Soit un système composé de deux processus cycliques Exécution et Impression, et de un tampon AVIS de N cases. Le tampon AVIS est géré de façon circulaire.

- Le processus Exécution exécute une requête de travail et transmet ensuite au processus Impression un ordre d'impression de résultats déposé dans le tampon Avis. La case de dépôt est repérée par un index i de dépôt, initialisé à 1. Les dépôts s'effectuent depuis la case 1 jusqu'à la case N, puis retour à la case 1 (gestion circulaire)
- Le processus Impression prélève les ordres d'impression déposés dans le tampon Avis et exécute ceux-ci.. La case de retrait est repérée par un index j de retrait, initialisé à 1. Les retraits s'effectuent depuis la case 1 jusqu'à la case N, puis retour à la case 1 (gestion circulaire)

**Question 1**

Programmez la synchronisation des deux processus à l'aide des sémaphores et des variables nécessaires à la gestion des tampons

Question 2

On étend la système à trois processus Exécution et trois processus Impression. Complétez la synchronisation précédente pour que celle-ci demeure correcte.

Exercice 3

On considère trois processus P1, P2, P3 dont les caractéristiques sont les suivantes

	Date d'arrivée	Temps d'exécution	priorité (plus petite valeur = plus grande priorité)
P1	0	6 unités	4
P2	2	4 unités	1
P3	4	7 unités	2

Les trois processus sont ordonnancés selon une politique de priorités préemptives.
On suppose à présent que les processus P1 et P2 utilisent une même ressource critique R1. P3 lui ne fait que du calcul.

Les étapes des deux processus P1 et P2 sont les suivantes :

P1	P2
Calcul durant 1 unité	Calcul durant 1 unité
Prendre(R1)	Prendre(R1)
Utiliser R1 durant 3 unités	Utiliser R1 durant 2 unités
Rendre (R1)	Rendre (R1)
Calcul durant 2 unités	Calcul durant 1 unité

Question 1

R1 est une ressource critique. Traduisez les opérations Prendre(R1) et Rendre(R1) à l'aide d'un sémaphore.

Question 2

Construisez le chronogramme d'exécution des trois processus en tenant compte du partage de la ressource R1 entre P1 et P2. Que se passe-t-il ? Est-ce que P2 s'exécute effectivement comme étant le processus le plus prioritaire ? Pourquoi ?