

Examen semestriel

Corrigé

Module : Systèmes d'exploitation I

Exercice 1 : Dans un système, on impose le protocole suivant : chaque processus qui demande une ressource supplémentaire doit d'abord libérer les ressources qu'il possède déjà.

Question 1 : Pensez-vous qu'on peut avoir un interblocage en exécutant ce protocole ? Justifiez.

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 d'"occupation et attente".

(2.5 points)

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

Réponse :

Si elle a l'avantage d'éviter les interblocages, cette méthode souffre du problème suivant : la durée nécessaire pour satisfaire tous les besoins des processus est allongée. On peut même avoir un problème de famine. Exemple : supposons qu'un processus P détient une ressource A, une ressource B et une ressource C, et supposons que ce processus demande une ressource supplémentaire B et une ressource D . Le protocole impose que P libère d'abord les ressources A, B et C. Ce qui veut dire que P, à qui manquait uniquement les ressources (1 B et 1 D), est devenu maintenant en quête des ressources suivantes : 1A, 2B, 1C et 1D , comme s'il venait de commencer.

(2.5 points)

Exercice 2 : Dans un système, on numérote arbitrairement les types de ressources de 1 à N (exemple : bande magnétique =1, processeur =2, imprimante=3, ...). On impose que chaque processus demande les ressources dont il a besoin dans l'ordre numérique croissant des ressources. Par exemple, un processus qui a besoin d'une imprimante et d'un processeur, doit d'abord demander le processeur ($n^{\circ}2$) et ensuite l'imprimante ($n^{\circ}3$).

Question 1 : Démontrez qu'avec cette démarche, il n'ya aucun risque d'interblocage

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 P_0, P_1, \dots, P_{n-1} , qui sont liés par une chaîne d'attente circulaire. Le processus P_0 possédant une ressource R_0 est en attente d'au moins une ressource R_1 de P_1 (avec $n^{\circ} R_0 > n^{\circ} R_1$ pour respecter la méthode), le processus P_1 est au moins en attente d'une ressource R_2 de P_2 (avec $n^{\circ} R_1 > n^{\circ} R_2$), et ainsi de suite ... jusqu'au processus P_{n-1} possédant une ressource R_{n-1} et attendant une ressource R_0 détenue par le processus P_0 (avec $n^{\circ} R_0 > n^{\circ} R_{n-1}$). D'où une contradiction. L'hypothèse de départ étant fautive, on n'a pas donc d'interblocage

(2.5 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.

(2.5 points)

Exercice 3 : Expliquez brièvement le principe, et donnez les avantages et les inconvénients des méthodes de communications suivantes : mémoire partagée, file, pipe et socket.

Réponse :

Méthode	Principe	Avantages	Inconvénients
Mémoire partagée	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.	La mémoire partagée est la forme de communication interprocessus la plus rapide car tous les processus partagent la même mémoire. Elle évite également les copies de données inutiles.	Le principe de l'exclusion mutuelle, nécessaire en cas d'accès concurrent, n'est pas garanti avec ce mode communication.
File	Une file premier entré, premier sorti (first-in, first-out, FIFO) est un tube qui dispose d'un nom dans le système de fichiers. Tout processus peut ouvrir ou fermer la file FIFO.	Les processus communicants peuvent ne pas avoir de lien de parenté.	Le principe de l'exclusion mutuelle, nécessaire en cas d'accès concurrent, n'est pas garanti avec ce mode communication.
Pipe	Un tube est un dispositif de communication qui permet une communication à sens unique. Les données écrites sur l'« extrémité d'écriture » du tube sont lues depuis l'« extrémité de lecture ». Les tubes sont des dispositifs séquentiels; les données sont toujours lues dans l'ordre où elles ont été écrites. Typiquement, un tube est utilisé pour la communication entre deux threads d'un même processus ou entre processus père et fils.	La synchronisation des processus en entrée et en sortie du tube est assurée par le dispositif lui-même.	Exigent que les processus soient liés par un lien de parenté (père-fils). Sens unidirectionnel.
Socket	Un socket est un dispositif de communication bidirectionnel pouvant être utilisé pour communiquer avec un autre processus sur la même machine ou avec un processus s'exécutant sur d'autres machines.	Possibilité de choisir le protocole, type de connexion et style de datagramme	Plus difficile de gérer plusieurs processus communicants en même temps

(5 points)

Exercice 4 : Sur un ordinateur, on dispose d'une instruction matérielle "Echanger" qui permet d'échanger de manière indivisible le contenu de deux variables. Son code est le suivant :

```
Procédure Echanger (var a, b : Logique)
Début

    Temp : Logique ;
    Temp :=a ;
    A :=b ;
    B :=Temp

Fin.
```

Question 1 : Montrer comment on peut réaliser l'exclusion mutuelle en utilisant cette instruction

Réponse :

On peut réaliser l'exclusion mutuelle en utilisant l'instruction Echanger, en déclarant une variable logique Lock initialisée à faux et une variable logique locale Key au niveau de chaque processus (voir figure suivante).

```
Processus Pi
Début
    Key :=Vrai ;
    Répéter Echanger(Lock, Key)
    Jusqu'à Key=Faux ;

    |
    | SC
    |

    Lock :=Faux
Fin ;
```

(4 points)

Question 2 : Quel est l'inconvénient de ce type de solution.

Réponse :

Les solutions matérielles du problème de la mutex sont des solutions spécifiques pour certains ordinateurs. Elles ne peuvent pas être appliquées dans tous les cas.

(1 points)