

Les exercices 3A et 3B sont au choix

Exercice 3A (Barème 6pts: Q1: 0,5pt – Q2: 1,5pt – Q3: 2,5pts – Q4: 1,5pt)

Dans le SGF d'UNIX, nous supposons une version modifiée de l'adressage avec un adressage indexé à plusieurs niveaux, des blocs disque de taille 400 octets et des adresses disque sur 4 octets. Sur cette structure, nous avons les adresses :

- a0 qui peut pointer vers 10 blocs contigus (certains de ces blocs peuvent être libres);
- a1 qui pointe vers une table d'indexées à 1 niveau;
- a2 qui pointe vers la racine d'une table d'indexées à deux niveaux.

1. Donner la taille maximale d'un fichier dans ce cas.
2. Calculer la taille maximale de l'espace réservé à l'adressage. Critiquer le résultat obtenu. Proposer une solution pour réduire cet espace.

Nous supposons un système de gestion de fichiers avec buffer cache.

3. Donner les différents accès effectués pour accéder au bloc 1207. Représenter ces accès sur un schéma de la structure.
4. Quelle serait la taille minimale du buffer cache pour avoir un nombre d'accès disque optimal dans le cas d'un fichier de taille 1530 blocs?

Exercice 3B: (Barème 6pts: Q1: 2pts – Q2: 3pts – Q3: 1pt)

Soit un état E défini par (ALLOC, DEMANDE, DISPO).

L'algorithme suivant permet de détecter l'interblocage à l'état E :

Algorithme

S,T : ensembles d'identités de processus;

i,j : entier ;

etat_sain : booléen ;

debut

```
S := ∅ ; etat_sain := faux ;
T := P ; /* P est l'ensemble des identités des processus du système supposés déjà initialisés */
```

Tant que ($T \neq \emptyset$) faire

i := indice d'un élément de T ;

T := T - {pi} /* pi ∈ P */

Si (DEMANDE_{pi} - ALLOC_{pi} <= DISPO + $\sum_{pj \in S}$ ALLOC_{pj}) alors

 pj ∈ S

 Début S := S + {pi} ;

 T := P - S

 Fin ;

 fst

 Fait ;

 Si (S = P) alors etat_sain := vrai ; fsi

Fin.

- 1- Expliquer brièvement le fonctionnement de cet algorithme.

A chaque exécution d'une nouvelle demande de ressources par un processus pi, en partant d'un état sain, l'état obtenu n'est pas forcément sain. L'interblocage peut donc apparaître. Il est donc nécessaire de faire une détection à chaque requête (on parle de détection continue).

Soit le théorème suivant : "L'exécution d'une requête par un processus pi fait passer le système d'un état sain à un nouvel état sain, s'il existe une suite saine (pas forcément complète) dans le nouvel état contenant ce processus."

- 2- En utilisant ce théorème, modifier l'algorithme donné pour faire une détection continue de l'interblocage.
- 3- Quel est l'apport de ce théorème sur la détection continue ?



Concours d'accès au Doctorat LMD Informatique 2015 – 2016

18-10-2015

Epreuve commune: Système d'exploitation

Durée: 1h30

Exercice 1 (Barème 7pts: QA.1: 1pt – QA.2: 1pt – QA.3: 1pt – QB.1: 3pts – QB.2: 1pt)

A- Répondre aux questions suivantes:

1. Quelle est la différence entre une ressource critique et une section critique?
2. Qu'apportent les sémaphores par rapport aux variables d'état dans la synchronisation ?
3. Un processus Pi s'exécute en section critique et un processus Pj, concurrent à Pi, vient d'être élu par l'ordonnanceur (politique Round Robin) et veut accéder à la section critique.
Donner l'état de Pj quand il trouvera que Pi est en section critique dans le cas d'une synchronisation en **Attente active** et une synchronisation avec **sémaphores**.

B- Soit l'algorithme suivant pour deux processus i et j:

```
Var D : tableau[1..2]de booléen := faux ;  
    Tour : entier := Valeur arbitraire i ou j;  
Processus Pi ;  
Debut  
    Repeter  
        Di :=Vrai ;  
        Tq (Dj et (tour<>i)) faire rien Fait ;  
        <SC>  
        Dj :=Faux ;  
        Tour :=i ;  
        Jusqu'à Faux ;  
Fin
```

1. Peut-on retenir ce protocole comme solution au problème de l'exclusion mutuelle ? Justifier.
2. Si on modifie la boucle comme suit : **Tq (Dj ou (tour=*)) faire rien Fait ;**
Que peut-on conclure pour le cas de deux processus ?

Exercice 2 (Barème 7pts: Q1: 2pts – Q2: 5pts)

Une illustration du problème de la synchronisation est celui d'une salle d'impression pendant la période d'examens. La salle d'impression dispose d'un seul agent, d'une seule machine pour l'impression des sujets d'examen et de 3 sièges pour l'attente.

Soient les hypothèses suivantes :

- Durant la journée, l'agent de salle est soit en train d'imprimer un sujet d'examen, soit en attente de l'arrivée d'un enseignant pour lui imprimer le sujet. L'agent ne peut faire qu'une seule impression à la fois.
 - Lorsqu'un enseignant se présente à la salle pour imprimer son sujet, deux scénarios se présentent à lui:
 - o Si l'agent de salle est libre, il procède à l'impression du sujet;
 - o Si l'agent de salle est occupé par l'impression d'un autre sujet alors:
 - Si l'enseignant trouve une chaise libre, il prend la chaise et attend son tour;
 - Sinon l'enseignant repart à son bureau.
 - Quand l'agent de salle termine l'impression d'un sujet, il vérifie si d'autres enseignants attendent. Dans ce cas, il invite le prochain enseignant pour lui imprimer le sujet, sinon l'agent se repose.
- 1- Identifier les ressources impliquées dans la synchronisation et déduire les sémaphores à utiliser et leur rôle.
 - 2- Synchroniser les deux processus à l'aide des sémaphores.