Université Bordeaux I
Session de janvier 1998

U.F.R. Mathématiques et Informatique
Alain Griffault

Maîtrise d'Informatique
Systèmes d'exploitation
(Durée : 2 heures)

Documents interdits

Avertissement

L'exercice 1 est court, mais demande de l'attention.
L'exercice 2 est assez long, mais facile.
L'exercice 3 est assez long. La difficulté consiste à interpréter dans un autre contexte, vos connaissances en SE.
L'ensemble est long, la notation en tiendra compte. (Le barème est indicatif de l'importance respective des exercices).

1 Synchronisation avec des moniteurs : 5 points

Définition : Le type moniteur est une structure de synchronisation de haut niveau, ayant la même puissance de synchronisation que les sémaphores. Un moniteur comprend des variables, des conditions, des procédures et un code d'initialisation. Seules les procédures peuvent être manipulées par les processus, et le mécanisme du moniteur garantit l'exécution de chaque procédure en exclusion mutuelle.

Syntaxe d'un moniteur

type *nom du moniteur* = monitor
    déclarations de variables
    déclarations de conditions
procedure entry *P1(...)*
    begin ... end ;
⋮
procedure entry *Pn(...)*
    begin ... end ;
begin
code d'initialisation
end.

La synchronisation des processus est obtenu à l'aide des procédures *wait* et *signal* appliquées sur les variables de type condition. L'action X.signal provoque le réveil de tous les processus qui ont auparavant effectué X.wait.

Une mauvaise solution au problème de l'exclusion mutuelle

```
type SectionCritique = monitor
        token : booleen ;
        sc : condition
procedure entry EntreeSC
        begin
        if token then token := false
        else sc.wait ;
        end ;
procedure entry SortieSC
        begin
        token := true
        sc.signal ;
        end ;
begin
token := true
end.

shared var SC = SectionCritique
```

Code des processus

```
begin
while true do begin
        code de la section non critique
        SC.EntreeSC ;
        code de la section critique
        SC.SortieSC
        end
end.
```

Question 1.1 *Une erreur classique en synchronisation de processus s'est glissée dans la solution proposée. Donnez brièvement :*

1. *L'erreur.*
2. *La condition au problème de l'exclusion mutuelle qui n'est pas satisfaite.*
3. *Une correction.*

Question 1.2 *Donnez le code des deux procédures* dormir *et* top *du moniteur Reveil suivant, afin qu'un processus puisse se retarder d'un nombre déterminé d'unité de temps. Vous pouvez supposer qu'une horloge matérielle, appelle à intervalle régulier la procédure* top.

Remarque : Des solutions, correctes dans un certain contexte, pourront être considérées comme bonnes si le contexte restrictif d'utilisation est mentionné.

Le moniteur

```
type Reveil = monitor
        déclarations de variables
        déclarations de conditions
procedure entry dormir(...)
        begin ... end;
⋮
procedure entry top(...)
        begin ... end;
begin
code d'initialisation
end.
```

shared var TicTac = Reveil;

Code des processus

```
var n : integer
begin
while true do begin
        code
        TicTac.dormir(n);
        end
end.
```

Code de l'horloge matérielle

```
begin
while true do begin
        code
        TicTac.top;
        end
end.
```

2 Algorithmes de remplacement de pages : 5 points

L'algorithme de la seconde chance est à la base un algorithme de remplacement FIFO, auquel on ajoute un mécanisme pour éventuellement donner un deuxième chance à la page la plus ancienne. Une variante de l'algorithme de la seconde chance utilise les deux bits de référence et de modification. Le bit de modification (1 pour modifiée, 0 sinon) est associé à chaque demande de page et le bit de référence (1 pour récemment utilisée, 0 sinon) est manipulé de la façon suivante :

- Mis à 1 au chargement de la page.
- Mis à 1 à chaque nouvel accès de la page.
- Périodiquement mis à 0 par le système.
- *Suivant l'implémentation choisi, éventuellement mis à 0 par l'algorithme de la seconde chance.*

Les quatre configurations possibles sont assimilables à des niveaux de priorité (P0, P1, P2 et P3).

Question 2.1 *Si P0 correspond au couple (0,0), donnez les deux seules associations réalistes (pour un algorithme de remplacement de pages) entre les niveaux (P0, P1, P2 et P3) et les quatre couples (bit_référence, bit_modification).*

Question 2.2 *Proposez un schéma de l'algorithme de la seconde chance qui utilise les notations (P0, P1, P2 et P3).*

Soit la suite de référence suivante, avec une indication supplémentaire indiquant si la page est modifiée. (L pour Lecture, E pour Ecriture), et des "top" indiquant les mises à 0 des bits de références par le SE.

(1,L), (2,E), top, (3,L), (4,L), (3,E), (1,L), top, (3,L), (2,E), top, (5,L), (4,L), (3,E)

Question 2.3 *Donnez pour chacune des deux associations possibles la suite des configurations en supposant que la mémoire est constituée de 3 cadres initialement vides.*

Question 2.4 *A partir de cet exemple (en comparant le nombre de défauts de page, et le nombre d'écritures forcées sur disque lors d'un remplacement de page), pouvez vous conclure sur les qualités respectives des deux associations.*

3 Interblocages : 10 points

Soit un ensemble de processus et un ensemble de ressources. Il y a un interblocage dans ce système si et seulement si toutes les conditions suivantes sont satisfaites.

Exclusion mutuelle les ressources ne sont pas partageables.

Occupation et attente un processus qui possède déjà une ressource n'est pas obligé de la libérer avant d'en prendre une autre.

Pas de réquisition une ressource ne peut être retirée à un processus pour être allouée à un autre.

Attente circulaire il existe un cycle dans le graphe d'allocation de ressources.

Imaginez la situation de la circulation présentée dans la figure 1. Pour simplifier, vous supposerez que les voies sont toutes à sens unique, et que les véhicules vont tout droit.

Question 3.1 *Après avoir précisé les ressources et les processus, montrez que les quatre conditions nécessaires pour l'interblocage sont réellement vérifiées dans cette situation.*

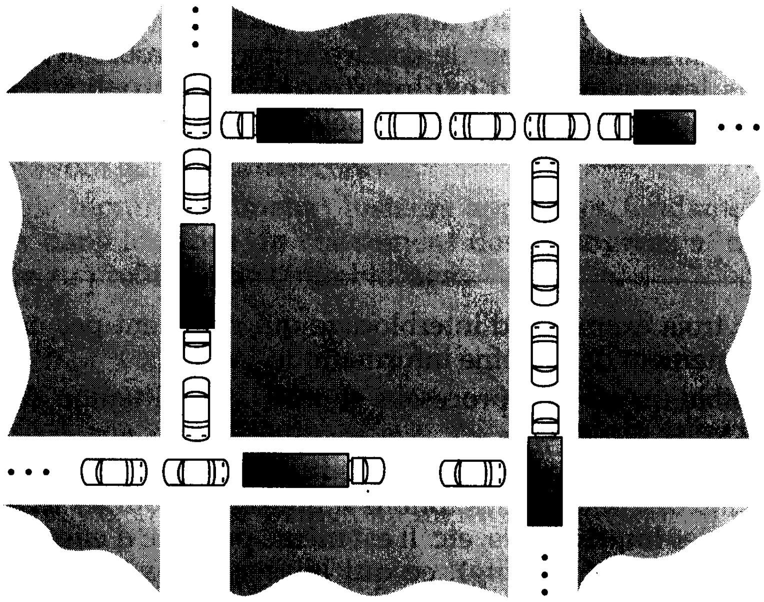

FIG. 1: un problème de circulation.

Le problème de l'interblocage peut être résolu de plusieurs façons :

- Par **prévention** en veillant à ce qu'une des quatre conditions soit toujours fausse.
- Par **évitement** en veillant à ce que le système soit toujours dans un état sain, (définition vue en cours et TD).
- Par **correction** après détection d'un interblocage.

Question 3.2 *Pour chacune des trois méthodes, donnez si possible une règle correspondante pour notre problème de circulation.*

Dans le code de la route français, il est dit qu'un conducteur ne peut s'engager dans un carrefour, que s'il est certain de ne pas y rester bloqué.

Question 3.3 *Expliquez brièvement sur quelle méthode est basé le code de la route français ?*

Question 3.4 *Le respect du code de la route transforme le problème de l'interblocage en un autre : lequel ?*
Donnez un exemple de situation dans laquelle ce nouveau problème est présent.
Dites pourquoi le code de la route (malgré ce problème) est sans doute une solution acceptable.

Question 3.5 *Proposez une solution au problème de l'interblocage pour la circulation routière. (Cette solution peut-être nouvelle, ou bien basée sur vos connaissances en systèmes d'exploitation).*