Nom : Prénom : Option : IEA IOE

IUP MIAGe 3^{eme} année — Année universitaire 95-96

Examen Programmation Parallèle — Alain Griffault

Mardi 22 mai, 14 heures - 16 heures
Tous documents autorisés

Exercice 1 (5 points) : Partage de ressources avec des sémaphores

Une bibliothèque contient une seule œuvre, composée de trois volumes. Tous les volumes sont disponibles en :

- P exemplaires accessibles d'une part par les lecteurs ; d'autre part par le bibliothéquaire.
- 1 exemplaire original accessible d'une part par l'auteur, d'autre part par le bibliothéquaire.

Une œuvre (les 3 volumes) est donc régulièrement : soit lue par un lecteur quelconque, soit modifiée par son auteur, soit mise à jour par le bibliothéquaire.
En terme d'algorithmes :

```
Algorithme de l'auteur
begin
while true do
  begin
  (* modifier l'original *)
  (* les 3 volumes *)
  end
end
```

```
Algorithme du bibliothéquaire
begin
while true do
  begin
  (* Mettre à jour les copies *)
  (* à partir de l'original *)
  end
end
```

```
Algorithme d'un lecteur
begin
while true do
  begin
  (* consulter une copie complète *)
  (* les volumes sont pris *)
  (* dans un ordre quelconque *)
  end
end
```

Modifiez ces algorithmes par l'ajoùt de sémaphores afin de garantir les deux contraintes suivantes :

- Deux lecteurs simultanés ont toujours des copies identiques.
- Une copie est toujours cohérente (ie, les volumes correspondent à une même version)

Pour chacun des sémaphores utilisés, décrivez brièvement son rôle.

Exercice 2 : Le problème des philosophes

Ce problème a été présenté en cours. La version donnée ici est légèrement différente, car les fourchettes ne sont pas identifiées, mais forment un ensemble.

```
[
Philosophe(i : 1..N)::
 *[ true --> penser
             Fourchettes ! prendre
             Fourchettes ! prendre
             manger
             Fourchettes ! poser
             Fourchettes ! poser ]
//
```

```
Fourchettes::
 Nombre := N
 *[ Philosophe(i) ? prendre and Nombre > 0
         --> Nombre := Nombre-1
    ◇
    Philosophe(i) ? poser
         --> Nombre := Nombre+1 ]
]
```

Question 1 (2 points)
Montrez, en donnant une exécution possible de ce programme pour N = 3, que cette solution n'est pas exempte de blocage. Un blocage étant une situation dans laquelle aucun des processus peut évoluer.

Instant	Philosophe 1	Philosophe 2	Philosophe 3	Fourchettes	Nombre
0 (initial)					3
1					
2					
3					
4					
5					
6					

Question 2 (5 points)
Montrez, en donnant une exécution possible de ce programme pour N = 4, que cette solution n'est pas exempte de famine. La famine est possible si plusieurs philosophes, en se liguant, peuvent empêcher indéfiniment un autre philosophe de manger.

Instant	Philosophe 1	Philosophe 2	Philosophe 3	Philosophe 4	Fourchettes	Nombre
0 (initial)	penser	penser	penser	penser		4
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						

Exercice 3 : Modélisation de feux tricolores.

Ecrivez en OCCAM ou en CSP les réponses aux questions.
On désire gérer la circulation automobile française d'un carrefour à l'aide de feux tricolores.
Un feu tricolore simple est un objet composé de :

- un écran capable d'afficher répétitivement les trois couleurs vert, orange et rouge.
- un contrôleur logiciel.

Le contrôleur peut-etre représenté par le processus suivant :

```
contrôleur ::
delai := 30
couleur := vert
*[ couleur = vert AND TIME ? AFTER delai -->
          couleur := orange
          ecran ! couleur
          delai := 3
   ◇
   couleur = orange AND TIME ? AFTER delai -->
          couleur := rouge
          ecran ! couleur
          delai := 15
   ◇
   couleur = rouge AND TIME ? AFTER delai -->
          couleur := vert
          ecran ! couleur
          delai := 30
]
```

Pour comprendre le comportement de ce processus, il suffit de savoir que le garde `TIME ? AFTER delai` devient VRAI lorsque le delai est écoulé. Ainsi ce processus affiche le vert pendant 30 secondes, puis l'orange pendant 3 secondes, enfin le rouge pendant 15 secondes, puis recommence.

Question 1 (2 points)
Un feu tricolore complexe est un objet composé de :

- un écran capable d'afficher répétitivement les trois couleurs vert, orange et rouge.
- un bouton sur lequel un piéton peut appuyer dans le but de provoquer rapidement un passage du feu au rouge, afin de pouvoir traverser la rue. Une fois rouge, le feu doit le rester un minimum de 10 secondes pour garantir la sécurité du piéton.
- un contrôleur logiciel.

Donnez le code du contrôleur d'un tel feu tricolore.

Question 2 (4 points)
On désire aménager un carrefour simple (deux rues qui se croisent avec interdiction de tourner) à l'aide de 4 feux tricolores complexes, ayant un contrôleur unique.
Donnez le code du contrôleur d'un tel carrefour. Attention au délai de sécurité.

Question 3 (6 points)
On désire aménager un carrefour à l'australienne (deux rues qui se croisent avec la possibilité de tourner à gauche sans croiser de véhicule) à l'aide de 4 feux tricolores simples et de 4 feux complexes, l'ensemble ayant un contrôleur unique.
Donnez le code du contrôleur d'un tel carrefour. Attention au délai de sécurité. Il peut être nécessaire que le piéton s'arrête au milieu de la rue.