

# Variante de l'algorithme d'exclusion mutuelle de Lamport

## TD6 et 7 - Algorithmique distribuée - Polytech Grenoble INFO4

L'algorithme suivant, dit de Ricart et Agrawala, résout le même problème que l'algorithme d'exclusion mutuelle de Lamport, selon les mêmes hypothèses. C'est une amélioration qui utilise un type de message en moins (message de sortie). Le but reste le même : attribuer la section critique à la demande minimale selon l'ordre total construit à partir des horloges logiques des processus et de l'ordre entre processus. Mais le principe diffère un peu : un processus, quand il requiert la ressource, demande l'autorisation aux autres processus et n'accède à la section critique que quand il a reçu l'autorisation de tous les autres processus.

**Question 1** – Implémenter dans SINALGO l'algorithme suivant. On pourra s'aider du squelette de code fourni ici : <http://www-verimag.imag.fr/~altisen/APD/> Tester plusieurs scénarios pour bien comprendre l'algorithme.

**Question 2** – En particulier, considérer 4 processus, 3 demandes selon les deux scénarios suivants

1. Un processus demande, puis un deuxième, puis le premier (une fois sa section critique exécutée) redemande.
2. Un processus demande, puis deux processus demandent “en même temps” (leurs horloges logiques doivent être égales).

Dessiner des diagrammes temporels illustrant ces scénarios. A noter : des éléments restent à préciser dans ces scénarios (ordre sur les processus, vitesse d'acheminement des messages) qui peuvent donner lieu à plusieurs sous-cas. Il ne s'agit pas ici de les étudier tous mais de comprendre l'algorithme en manipulant des exemples.

**Question 3** – Prouver que *la section critique est toujours utilisée en exclusion mutuelle*.

**Question 4** – Démontrer que *toute demande est servie en temps fini*.

**Question 5** – Puis conclure quant à la correction de l'algorithme.

**Question 6** – Quel est le temps de service de l'algorithme (c'est-à-dire le nombre maximum de processus pouvant être servis avant qu'un processus donné ne le soit à son tour) ?

**Question 7** – Combien y a-t-il de messages échangés par demande servie ?

**Question 8** – Comparer cette solution avec l'algorithme de Lamport.

---

**Algorithme 1** Exclusion mutuelle sur une clique, algorithme de Ricart et Agrawala pour un processus  $p$

---

**Variables**

- 1:  $Demandeur \in \{Vrai, Faux\}$
- 2:  $dd$  date de la demande de  $p$
- 3:  $h$  horloge logique de Lamport pour  $p$
- 4:  $DR$  tableau de booléens de demandes retardées,  $DR[q]$  signifie que le processus  $p$  retarde la demande du processus  $q$
- 5:  $nb\_perm$  nombre de permissions reçues
- 6: on note  $(q, d_q) \ll (r, d_r)$  pour signifier que la demande du processus  $q$  à la date  $d_q$  a eu lieu avant (selon l'ordre total) la demande du processus  $r$  à la date  $d_r$ , c'est-à-dire,  $d_q < d_r$  ou  $d_q = d_r$  et  $q < r$ .

**Algorithme pour tout processus  $p$**

```
7:  $Demandeur \leftarrow Faux$ 
8:  $h \leftarrow 0$ 
9:  $DR \leftarrow \{Faux, \dots, Faux\}$ 
10: Pour toujours
11:   Réceptionner  $\langle Type, t \rangle$  de  $q$ 
12:    $h \leftarrow \max(h, t + 1)$ 
13:   Si  $Type = Demande$  alors
14:     Si  $Demandeur = Vrai$  et  $(p, dd) \ll (q, t)$  alors
15:        $DR[q] \leftarrow Vrai$ 
16:     Sinon
17:       Envoyer  $\langle Permission, h \rangle$  à  $q$ 
18:     Fin Si
19:   SinonSi  $Type = Permission$  alors
20:      $nb\_perm \leftarrow nb\_perm + 1$ 
21:     Si  $nb\_perm = n - 1$  alors
22:       SC ▷ Section critique
23:        $Demandeur \leftarrow Faux$ 
24:       Pour tout processu  $q \neq p$  faire
25:         Si  $DR[q] = Vrai$  alors
26:           Envoyer  $\langle Permission, h \rangle$  à  $q$ 
27:            $DR[q] \leftarrow Faux$ 
28:         Fin Si
29:       Fin Pour
30:     Fin Si
31:   Fin Si
32:    $h \leftarrow h + 1$ 
33: Fin Pour toujours
```

**Sur demande**

```
34: Si  $Demandeur = Faux$  alors
35:   Envoyer  $\langle Demande, h \rangle$  à tous les voisins
36:    $dd \leftarrow h$ 
37:    $Demandeur \leftarrow Vrai$ 
38:    $nb\_perm \leftarrow 0$ 
39:    $h \leftarrow h + 1$ 
40: Fin Si
```

---