

Algorithme de Carvalho Roucairol

Sur un SD de 4 sites. Le scénario :

- 1- 1 demande la SC
- 2- 2 demande la SC
- 3- 1 et 2 demandent la SC

Ne pas oublier d'**initialiser** (au hasard) puis dérouler l'algorithme.

(Se déroule sur un autre site, le site j)

```

25: /* Réception message(dem, (h', j)) de j */
26: h ← max(h, h')
27: priorite ← (etat = SC) ∨ [(etat = E) ∧ (last, i) < (h', j)]
28: si priorite = vrai alors
29:   differe ← differe ∪ {j}
30: sinon
31:   Envoi message(perm(i, j)) à j
32:   R ← R ∪ {j}
33:   si etat = E alors
34:     Envoi message(dem, (last, i)) à j
35:   fin si
36: fin si
  
```

```

1: R ensemble de sites tel que i ne possède pas perm(i, j), initialisé à {j | perm(i, j) absent de i}
2: etat ∈ {E, SC, S} etat du site i, initialisé à S
3: h entier, date des demandes, initialisé à 0
4: last entier, date de la dernière demande, initialisé à 0
5: differe ensemble de sites, qui retardent l'envoi d'une permission, initialisé à ∅
6: priorite boolean, indique si i prioritaire ou non, initialisé à faux

7: /* Demande d'entrée en section critique */
8: etat ← E
9: last ← h + 1
10: pour tout j ∈ R faire
11:   Envoi message(dem(last, i)) à j
12: fin pour
13: tant que R ≠ ∅ faire
14:   Reception message(perm(i, j)) de j
15:   R ← R \ {j}
16: fin tant que
17: etat ← SC

18: /* Sortie */
19: etat ← S
20: pour tout j ∈ differe faire
21:   Envoi message(perm(i, j)) à j
22: fin pour
23: R ← differe
24: differe ← ∅
  
```

i le site qui exécute cet algorithme, lui même

Temps où le site est en SC puis quand il a terminé ...

Dans cet algorithme un site conserve les permissions qui lui sont accordées tant qu'elles ne lui sont pas réclamées. Avantage : réduction du nombre de messages échangés pour chaque entrée en section critique (**de 0 à 2n - 2**).

Initialisation faite au hasard :

Entre ... et ...		Celui qui a la permission est ...
1	2	2
1	3	1
1	4	4
2	3	3
2	4	2
3	4	3

R est l'ensemble de sites dont i **n'a pas la permission**. Par exemple 1 à la permission sur 3 mais pas sur 2 et 4. Donc R = {2,4} lors de l'initialisation de l'algorithme. Suivre le même principe pour chaque site.

Tableau récapitulatif :

	1	2	3	4
Etat	S E SC S E SC S	S E SC S E SC S	S	S
H	0 2	0 1 3	0 2	0 1
Last	0 1 3	0 2	0	0
R	{2,4} ∅ {2} ∅	{3} {1,3} ∅ {1}	{1} {1,2}	{2,3} {1,2,3}
Diffère	∅	∅ {1} ∅	∅	∅
Priorité	F	F V	F	F

Dans un premier temps on initialise tous les sites avec leur valeur de départ (--).

1- 1 demande la SC

Les valeurs sont (--). Le site 1 attend, ligne 11, l'accord de tous les sites dont il n'a pas la permission {2,4}. Il demande donc la permission à ces sites pour passer en section critique.

- 1 envoie la demande de message $dem(last = 1, i = 1)$ à 2
- 1 envoie la demande de message $dem(last = 1, i = 1)$ à 4

Puis on se place sur le site qui reçoit la demande et on exécute le code *Réception de message*. A noter que la ligne 27 est au **sens de l'horloge**. C'est-à-dire : **((last = h') ET (i < j))** ou sinon **(last < h')**

- Sur le site 2 :
 - o Ligne 26 : $max(h = 0, h' = 1) = 1$
 - o Ligne 27 : faux
 - o Ligne 31 : envoi du message de permission à 1
 - o Ligne 32 : la variable « R » prend la valeur 1 en plus de celle qu'elle avait déjà.
- Sur le site 4 :
 - o Ligne 26 : $max(h = 0, h' = 1) = 1$
 - o Ligne 27 : faux
 - o Ligne 31 : envoi du message de permission à 1
 - o Ligne 32 : la variable « R » prend la valeur 1 en plus de celles qu'elle avait déjà.

On se replace sur le site 1 qui avait fait les demandes de permission. Il reçoit donc les 2 accords qu'il lui manquait et peut sortir de la boucle tant que de la ligne 13. Il rentre donc en section critique pour une durée indéterminée. Une fois la section critique terminée il change son état ligne 19. La variable « R » prend la valeur de « diffère » qui est vide et « diffère » garde sa valeur vide.

2- 2 demande la SC

Les valeurs sont (--). Ne pas oublier de donner la nouvelle valeur de la variable « last » et de changer l'état du site 2 au début de l'exécution de l'algorithme. Puis, le site 2 attend, ligne 14, l'accord des sites dont il n'a pas la permission de base donc {1,3}. Il demande donc la permission à ces sites pour passer en section critique.

- 2 envoie la demande de message $dem(last = 2, i = 2)$ à 1
- 2 envoie la demande de message $dem(last = 2, i = 2)$ à 3

Puis on se place sur le site qui reçoit la demande et on exécute le code *Réception de message*. Il faut se rappeler que la variable « h » est le h du site où on exécute le code, pas celle de celui qui avait fait la demande.

- Sur le site 1 :
 - o Ligne 26 : $max(h = 0, h' = 2) = 2$
 - o Ligne 27 : faux
 - o Ligne 31 : envoi du message de permission à 2
 - o Ligne 32 : la variable « R » prend la valeur 2 puisqu'elle n'a plus la permission de 2
- Sur le site 3 :
 - o Ligne 26 : $max(h = 0, h' = 2) = 2$
 - o Ligne 27 : faux
 - o Ligne 31 : envoi du message de permission à 2
 - o Ligne 32 : la variable « R » prend la valeur 2 en plus de celle qu'elle avait déjà.

On se replace sur le site 2 qui avait fait les demandes de permission. Il reçoit donc les 2 accords qu'il lui manquait et peut sortir de la boucle tant que de la ligne 13. Il rentre donc en section critique pour une durée indéterminée. Une fois la section critique terminée il change son état ligne 19. La variable « R » prend la valeur de « diffère » qui est vide et « diffère » garde sa valeur vide.

3- 1 et 2 demandent la SC

Les valeurs sont (--). Le site 1 et 2 demandent la section critique en même temps. Mettre à jour les variables au début de l'algorithme.

Puis, le site 1 attend de son côté, ligne 14, l'accord du site {2}.

- 1 envoie la demande de message $dem(last = 3, i = 1)$ à 2

Le site 2 n'a pas à faire sa demande de section critique car pour le moment il a l'accord de tous les sites.

On se place sur le site 2 qui reçoit la demande du site 1 et on exécute le code *Réception de message*.

- Sur le site 2, demande du site 1 :
 - o Ligne 26 : $\max(h = 1, h' = 3) = 3$
 - o Ligne 27 : $(last = 2, i = 2) < (h' = 3, j = 1)$ donc vraie au sens de l'horloge
 - o Ligne 29 : rajouter dans la variable « diffère » {1}

Pendant ce temps le site 2 rentre en section critique puis en ressort ligne 19. En effet, il est prioritaire sur 1 puisque sa variable « priorité » est à vraie. Il rentre dans la boucle tant que, ligne 20, et envoie sa permission au site 1 qui attend encore pour passer en section critique. Il n'a donc plus la permission sur le site 1, $R = \{1\}$, et sa variable « diffère » reprend sa valeur initiale, vide.

Le site 1 reçoit donc la permission du site 2 et rentre en section critique avant d'en ressortir et de finir son exécution.