

Recueil d'exercices et problèmes solutionnés sur les algorithmes distribués

Partie : Problème de la section critique

Elaboré par Pr. Djamel Eddine SAIDOUNI

Laboratoire MISC, Département IFA, Faculté des NTIC, Université Constantine 2 – Abdelhamid Mehri
saidounid@hotmail.com

Exercice 1:

Donner une comparaison entre l'algorithme de Ricard et Agrawala, et celui de Carvalho et Roucairol résolvant chacun le problème de l'accès à la section critique dans un contexte distribué.

Corrigé :

Dans la comparaison on doit mentionner les points qui différencient les deux algorithmes.

	Ricard et Agrawala	Carvalho et Roucairol
Nombre de messages de contrôle	Nombre = $2*(n-1)$	$0 \leq \text{Nombre} \leq 2*(n-1)$
Temps d'accès à la SC lorsqu'elle est libre où T est le temps moyen de transit d'un message entre deux sites	Temps = $2 * T$	$0 \leq \text{Temps} \leq 2 * T$
Adaptativité	Non adaptatif	Adaptatif
Bornétude des variables	Variables bornées	Variables non bornées

Exercice 2

Remplissez le tableau récapitulatif résumant les propriétés de quelques algorithmes distribués résolvant le problème de la section critique pour un processus P_i . En se fixe les hypothèses suivantes

- **N** étant le nombre de sites.
- Sur chaque site il existe un seul processus.
- **T** est le temps moyen de transit d'un message entre deux sites.

Nom de l'alg.	Nombre Message requête	Nombre message permission	Nombre message retour permission	Temps d'entrée en S.C. lorsqu'elle est libre	Adaptatif
Ricart Agrawala					
Carvalo Roucairol					
Misra Chandi					
Alg. Mixte					

Corrigé :

Nom de l'alg.	Nombre Message requête	Nombre message permission	Nombre message retour permission	Temps d'entrée en S.C. lorsqu'elle est libre	Adaptatif
Ricart Agrawala	$N-1$	$N-1$	0	$2*T$	Non
Carvalo Roucairol	$0 \leq \text{nbr} \leq N-1$	$0 \leq \text{nbr} \leq N-1$	0	$0 \leq \text{temps} \leq 2*T$	Oui
Misra Chandi	$0 \leq \text{nbr} \leq N-1$	$0 \leq \text{nbr} \leq N-1$	0	$0 \leq \text{temps} \leq 2*T$	Oui

9. Montrez que l'expression qui calcule la priorité d'un site **i** recevant une requête de la part d'un site **j** est équivalente à :

($\text{étati} = \text{dedans}$) ou (($\text{étati} = \text{demandeur}$) et ($(\text{lasti}, i) < (\text{lastj}, j)$))

Corrigé :

1. Quelles sont les variables utilisées dans l'algorithme de Ricart et Agrawala ? donnez le rôle de chacune d'elles.

Variable	Rôle
$\text{etat}_i = \{\text{dehors}, \text{demandeur}, \text{dedans}\}$ initialisé à dehors	Indique l'état du processus Dehors : le processus n'est pas intéressé par la section critique Demandeur : le processus désire accéder à la section critique Dedans : le processus est en section critique
h_i : entier croissant initialisé 0	Horloge logique utilisée pour dater les événements
last_i : entier croissant initialisé 0	Mémoire la date de la dernière demande du processus i
attendu_i : ensemble des sites initialisé à \emptyset	Ensemble des processus à partir desquels le processus i attend des permissions
differe_i : ensemble des sites initialisé à \emptyset	Ensemble des processus dont le processus i a retardé l'envoi de permission
priorité_i : booléen	Utilisé pour calculer la priorité

2. Rappelez

2.1. Le protocole de mise à jour des horloges logiques.

Lors d'un appel à acquérir :

$$h_i = h_i + 1 ;$$

Lors de la réception d'une requête de la part d'un site j avec une estampille (k,j) :

$$h_i = \max (h_i , k) ;$$

2.2. Le contenu d'un message Requête :

Le message Requête contient :

last_i : la date d'émission de la requête

i : l'identité du site émetteur

2.3. L'expression qui calcule la priorité d'un site **i** recevant une requête de la part d'un site **j**.

$\text{priorité} = (\text{etat}_i \neq \text{dehors}) \text{ et } ((\text{last}_i, i) < (k, j))$

ou bien :

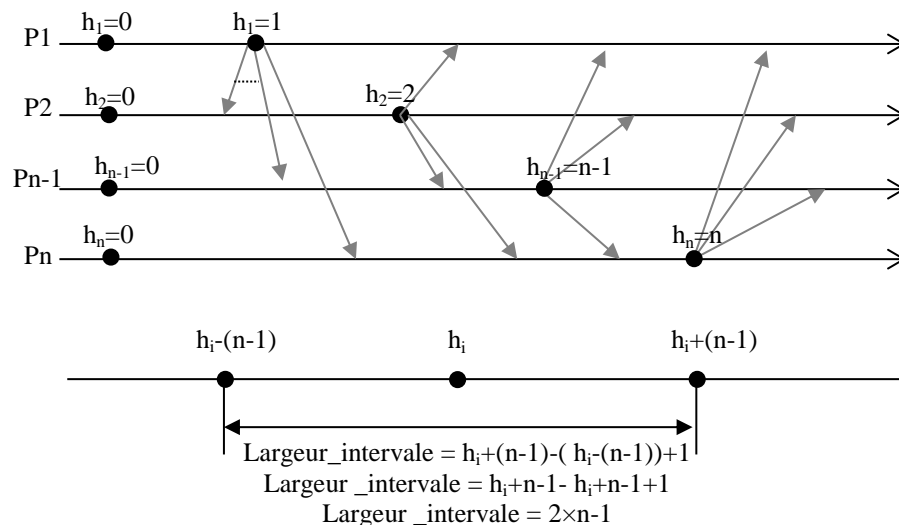
$\text{priorité} = (\text{etat}_i \neq \text{dehors}) \text{ et } ((\text{last}_i < k) \text{ ou } ((\text{last}_i == k) \text{ et } (i < j)))$

4. Peut-on borner les horloges logiques dans cet algorithme ? Justifiez votre réponse.

Oui, les horloges peuvent être bornées dans cet algorithme. En effet si h est la plus grande date associée à une requête, la prochaine date utilisée par une requête ne peut pas être supérieure à $h+1$. En supposant qu'il y a n sites et qu'un site ne peut pas faire une nouvelle requête que lorsque la précédente a été satisfaite, l'écart maximal entre les dates de deux requête est borné par $n-1$. Si l'on se place sur un site i , on sait donc que les autres horloges ont des valeurs dans l'intervalle :

$$[h_i-(n-1), h_i+(n-1)]$$

Les 2 bornes de l'intervalle correspondent aux cas extrêmes où h_i serait respectivement la plus grande et la plus petite valeur de l'horloge.



Etant donné que la largeur de l'intervalle est égale à $2 \times n - 1$, ainsi les horloges sont implémenté modulo $2 \times n - 1$.

5. Cet algorithme est de quel type ?

Cet algorithme appartient à la classe des algorithmes basés sur les permissions individuelles, puisque chaque site i gère individuellement chacun des conflits qui le mettent en compétition avec les autres sites.

6. Donnez les avantages et les inconvénients de cet algorithme.

Avantage :

- Les horloges bornées

Inconvénients :

- Non adaptatif puisque si le processus **i** est le seul demandeur, alors il doit envoyer ses requêtes aux autres sites (qui sont dans l'état dehors) et il attend leurs permissions, ainsi il y aura une perte de temps vu que la section critique est libre.
- Chaque site doit connaître les identités des autres sites

7. Citez un algorithme de même type qui surmonte les inconvénients de l'algorithme de Ricart et Agrawala sans pour autant conserver ses avantages. Expliquez son principe.

C'est l'algorithme de Carvalho et Roucairol puisqu'il est adaptatif mais il ne peut pas borner les horloges, son principe est le suivant :

- Associer à tout couple de site distinct **i** et **j** un et un seul message **permission(i,j)**.
- Si le message est chez **i**, celui-ci a la permission de **j** et inversement.
- Initialement **permission(i,j)** est placé indifféremment sur l'un ou l'autre site.
- Lorsqu'un site **i** invoque l'opération acquérir, il doit demander les permissions qui lui manquent, les sites qui détiennent ces permissions constituent donc l'ensemble **R_i**.
- Lorsque le site **i** reçoit **permission(i,k)**, il supprime **k** de **R_i**.
- Lorsque le site **i** envoie **permission(i,m)**, il rajoute **m** à **R_i** et dans le cas où le site **i** est dans l'état demandeur, il doit envoyer un message requête au site **m** afin de la récupérer une fois que le site **m** libérera la section critique.

L'application de ce principe permet de réduire le nombre de messages nécessaire à l'utilisation de la section critique. Par exemple si le processus **i** est le seul demandeur et il possède toutes les permissions alors il n'envoie pas de requêtes aux autres sites (qui sont dans l'état dehors) et il accède directement à la section critique vu qu'elle est libre. Cependant, il n'est pas possible de borner les horloges.

8. Citez un algorithme de même type qui surmonte les inconvénients de l'algorithme de Ricart et Agrawala tout en conservant ses avantages. Expliquez son principe.

C'est l'algorithme de Chandy et Misra puisqu'il est adaptatif (il utilise le même principe de Carvalho et Roucairol pour la distribution initiale des messages permissions entre les

différents sites) et il n'utilise pas des variables de taille bornée (horloges), son principe est le suivant :

- Associer un état à chaque permission. Ainsi un message permission peut être dans l'état *util* (utilisé) ou **non-util** (non utilisé).
- Initialement tout message permission est dans l'état util.
- L'état de ce message est mémorisé dans **permission(i,j).etat**.
- Lorsqu'un site **i** pénètre en section critique toutes ses permissions passent dans l'état **util**.
- Avant d'envoyer un message **permission(i,j)**, le site émetteur **i** passe l'état de ce message à **non-util**.
- Lorsque le site **i** reçoit un message requête du site **j**, est prioritaire sur **j** si le message **permission(i,j)** est dans l'état **non-util**, il diffère alors l'envoi de cette permission à sa sortie de la section critique. Dans le cas contraire, il envoie la permission par retour de courrier

Exercice 5

Dans l'algorithme de Ricard et Agrawala, un site **i** recevant une requête(**K,j**) calcule sa priorité par rapport à cette requête par la formule suivante :

$$\text{Priorité}_i := (\text{étati} \leq \text{dehors}) \text{ et } (\text{last}_i, i) < (K, j)$$

1. Montrez mathématiquement que cette formule est équivalente à la formule suivante :

$$\text{priorité}_i := (\text{étati} = \text{dedans}) \text{ ou } ((\text{étati} = \text{demandeur}) \text{ et } (\text{last}_i, i) < (K, j))$$

2. A l'aide d'un contre exemple, montrez que la première formule est incorrecte pour le calcul de la priorité dans l'algorithme de Carvalho et Roucairol. Quelle est la conséquence de l'utilisation de cette formule dans cet algorithme ? Justifiez vos réponses.

Corrigé :

Exercice 6

On se propose d'étudier la validité d'un algorithme résolvant le problème de la section critique dans un environnement distribué. On admet les hypothèses suivantes :

- **N** un entier non nul qui désigne le nombre de sites. Chaque entier **i** ($0 \leq i \leq N-1$) désigne un site qu'on note **Si**.
- Chaque site héberge un et un seul processus. De ce fait un processus est identifié par le site où il est hébergé et inversement.
- A chaque site **Si** sont associées :
 - Une horloge logique **Hi** initialisée à **0**.

- Une variable **Lasti** qui mémorise la date à laquelle le site est devenu demandeur.
- Une file d'attente **Filei** ordonnée, initialement vide. Cette file contient les estampilles des sites demandeurs d'accès à la section critique ayant sollicités la permission du site **Si**.
- **Laststampij** avec $0 \leq j \neq i \leq N-1$: Ensemble de variables. La variable Laststampij, initialisée à (0,j), mémorise la dernière estampille du message reçu par Si en provenance de Sj.

Le comportement d'un site (processus) Si est donné par les énoncés suivants :

Lors d'un appel à acquérir :

- $H_i := H_i + 1$
- $Last_i := H_i$
- Envoyer(Request($Last_i, i$)) à tous les autres sites
- Enfiler(($Last_i, i$)) dans Filei.
- Attendre (($Last_i, i$) en tête de Filei **et** (($Last_i, i$) < Laststampij pour tout $0 \leq j \neq i \leq N-1$))

Lors d'un appel à libérer :

- Défiler(($Last_i, i$)) de Filei
- Envoyer (Release((H_i, i))) à tous les autres sites

Lors de la réception de Request (K,j) :

- $H_i := \max(H_i, K)$
- Enfiler((K, j)) dans Filei
- $H_i := H_i + 1$
- Laststampij := (K, j)
- Envoyer(Response(H_i, i) à j

Lors de la réception de Response(K,j)

- $H_i := \max(H_i, K)$
- Laststampij := (K, j)

Lors de réception de Release(K,j)

- Défiler(l'estampille de Sj) de Filei

Questions :

Partie 1 : Compréhension de l'algorithme

1. L'exécution locale de chaque énoncé doit satisfaire une condition particulière, laquelle ? Justifier votre réponse.
2. Donner la fonction booléenne qui compare deux estampilles (L, i) et (K,j) où i et j sont les identificateurs des sites.
3. Les valeurs d'une variables Laststampij peuvent elle décroître dans le temps ? Justifier votre réponse.
4. A travers un scénario, montrer la **nécessité** de l'hypothèse de canaux FIFO pour la propriété d'exclusion mutuelle.

Partie 2 : Analyse de l'algorithme

5. Sous cette hypothèse, montrer que la propriété d'exclusion mutuelle est assurée par cet algorithme. (**indication** : Procéder par l'absurde en supposant qu'à un moment donné deux sites Si et Sj sont en section critique en même temps)
6. Montrez que la propriété de vivacité locale est assurée (chaque site demandeur arrivera à accéder à sa section critique). (**indication** : Procéder par l'absurde en supposant que Si et Sj

sont demandeurs d'accès à la section critique avec l'hypothèse que l'estampille de S_i est inférieure à l'estampille de S_j et que S_j peut rentrer à sa section critique avant S_i).

7. Dans le cas où un site tombe en panne, que ce passera-t-il ?
8. Calculer le nombre de messages mis en jeu pour chaque entrée en section critique.
9. Le nombre de messages mis en jeu peut être diminué par la diminution des messages « Réponse ». Donnez cette amélioration ainsi que le nouveau nombre de messages.

Partie 3 : Amélioration de l'algorithme

Un diplômé d'un master en informatique a constaté que pour son application les sites se divisent en deux catégories :

- **Catégorie1** : Sites qui demandent rarement l'accès à la section critique,
- **Catégorie2** : Sites qui demandent très souvent l'accès à la section critique.

De ce fait, il propose une modification de l'algorithme afin de diminuer le nombre de messages mis en jeu pour l'accès à la section critique.

Modification : En dehors d'une catégorie, seuls les processus de la catégorie2 sont sollicités par les processus demandeurs d'accès à la section critique.

10. Apporter les modifications nécessaires aux énoncés de l'algorithme.
11. Soit n_1 (respectivement n_2) le cardinal de Catégorie1 (respectivement Catégorie2).
 - a. Quel est le nombre de messages de contrôle mis en œuvre par un processus de Catégorie1 pour accéder à sa section critique.
 - b. Même question pour un processus de Catégorie2.
12. Soit T une période de temps. On suppose qu'à chaque période T , chaque site de Catégorie2 demande l'accès à sa section critique, alors que chaque site de Catégorie1 demande l'accès à chaque période $f \cdot T$ (f est un entier positif).
 - a. Dans la période T , quel est le nombre de messages engendrés par les processus de Catégorie2.
 - b. Dans la période $f \cdot T$, quel est le nombre de messages engendrés par les processus de Catégorie1.
 - c. Dans la période $f \cdot T$, quel est le nombre total de messages engendrés par tous les processus.
 - d. Discuter le cas où $f = 1$ et $n_1 = 0$.
13. La sûreté de l'algorithme est-elle préservée pour $n_1 = n_2 = 1$? Argumenter votre réponse.

Corrigé :

Partie 1 :

1. La condition particulière que chaque énoncé doit satisfaire est la propriété d'atomicité sauf pour l'instruction Attendre qui est interruptible.

Justification : Remarque par exemple que la variable locale H_i est manipulée en écriture dans les deux procédures concurrentes « Appel à acquérir » et « Réception d'une requête » de ce fait, cette variable doit être manipulée en EM.

2. La fonction booléenne qui compare deux estampilles, (L,i) et (K,j) est basée sur l'ordre lexicographique :
 $(L,i) < (K,j)$ ssi $(L < K)$ ou $(L=K \text{ et } i < j)$
3. Les valeurs de Laststampij ne vont que croître car les horloges des sites sont synchronisées à chaque événement et que **les liaisons sont FIFO**.
4. Sans l'hypothèse FIFO des liaisons, l'exclusion mutuelle est non garantie. Exemple : Soient les deux sites i et j respectivement avec $i < j$ demandeurs d'accès à la SC en même temps. Acceptant que la requête du site i arrive très en retard au site j (canaux non FIFO). j décide de rentrer dans sa SC étant donné qu'il est en tête de file j et que Response de i est arrivée. Après l'arrivée de Response de j chez i et comme l'estampille de i est inférieur donc i est en tête de sa file. Par conséquent il rentre dans sa SC.

Deuxième partie :

5. Procédons par l'absurde : Si et Sj en SC. Pour cela la condition de franchissement est vérifiée par i et j . A un instant t , les deux requêtes de S_i et S_j sont en tête de file i et file j respectivement. Supposons maintenant que $(Last_i,i) < (Last_j,j)$, à la réception de des messages Response de S_i par S_j et de S_j par S_i , la tête de file de j est $(Last_i,i)$ ce qui contredit l'hypothèse que $(Last_j,j)$ est aussi tête de file de S_j .
6. Supposons que $(Last_i,i) < (Last_j,j)$ et S_j exécute la Sc avant S_i . Pour que S_j exécute la Sc il doit être en tête de sa file d'attente et $Laststamp_j > (Last_j,j)$. Or $(Last_i,i) < (Last_j,j)$ et donc elle précède la requête de j dans file j , d'où la contradiction. La vivacité locale (individuelle) est vérifiée.
7. Interblocage.
8. $3*(N-1)$
9. Amélioration :

Enoncé Réception d'une requête :

Si $(Last_i,i)$ not supérieur Laststampij alors Envoyer Response (Hi,i) à j

$$2*(N-1) \leq \text{Nombre de message} \leq 3*(N-1)$$

Troisième partie :

Convention : Processus de Catégorie 2 numérotés de 0 à $n_2 - 1$

10. Enoncé Acquérir : (Juste les sites de la catégorie 2)
 Attendre $(Last_i,i) < Laststamp_{ij}$ pour $(0 \leq j < i \leq n_2 - 1)$
11. Réponse
 - a. $3*(N-1)$
 - b. $2*(N_2 - 1) + (N - 1)$
12. Réponse
 - a. $3 * (N_2 - 1)$ ou $N_2 * 3*(N_2 - 1)$ si on suppose que la durée de la Sc est négligeable
 - b. $3*(N - 1) + (N_1 - 1)$ Ou bien si on suppose que la durée de la Sc est négligeable $(N_1 * 3 * (N-1))$
 - c. $f*(11.a) + 11.b =$
 - d. Dans ce cas $N_2 = N$ et en retombe sur le premier algorithme

13. La sûreté n'est pas assurée (voir exemple)

Exercice 7 :

Soient les éléments de réponses suivants :

- a) En ce qui concerne les processus qui m'ont sollicité vous pouvez entrer en section critique.
- b) Principe de l'égalité de la responsabilité.
- c) En ce qui me concerne vous pouvez entrer en section critique.
- d) Principe de l'égalité de l'effort.

Pour un algorithme résolvant le problème de la section critique, choisissez la réponse qui convient :

- 1) Pour D_i = ensemble des processus qui sollicitent le processus P_i . La sémantique de $|D_i| = \text{Constante}$ est : a b c d
- 2) Si l'algorithme est à permission d'arbitre, la sémantique d'une permission donnée par un processus est : a b c d
- 3) Pour R_i = ensemble des processus sollicités par le processus P_i . La sémantique de $|R_i| = \text{Constante}$ est : a b c d
- 4) Si l'algorithme est à permission individuelle, la sémantique d'une permission donnée par un processus est : a b c d

Corrigé :

Exercice 7 :

Parmi les problèmes posés par les algorithmes basés sur les jetons et résolvant le problème de la section critique sur un anneau on trouve le problème de la perte du jeton et la nécessité de sa régénération. Dans cet exercice on se propose d'étudier un des algorithmes résolvant ce problème. On suppose l'existence de n processus sur n sites répartis sur un anneau.

Le principe de cet algorithme consiste en la mise en circulation de deux jetons ping et pong. Un site recevant l'un des jetons déclarera que l'autre jeton est perdu si le jeton reçu a fait un tour complet sans que le site en reçoive l'autre jeton entre temps. Dans le cas affirmatif, le site régénère le jeton perdu et le remet en circulation sur l'anneau.

- 1. En regardant les lignes L3 et L4 ainsi que les lignes L6 et L7 ; quelle est la valeur de $Nbping + Nbpong$? Que peut-on conclure ?
- 2. Que comptent $Nbping$ et $|Nbpong|$?
- 3. Au niveau de la ligne L1, si P_i trouve que $m = nbping$ il déclare que pong est perdu, expliquez pourquoi ?
- 4. Peut-on borner les variables $Nbping$ et $Nbpong$? Argumentez votre réponse.
- 5. Que ce passe t-il lorsque les liaisons ne sont pas fifo ?
- 6. Que ce passe t-il si les deux jetons se perdent ? Quelle solution proposeriez-vous pour remédier à ce problème ?

7. Expliquez comment cet algorithme est utilisé pour résoudre le problème de la section critique ?

L'algorithme :

Var m : entier initialisé à 0

Le comportement du processus P_i est le suivant :

Lors de

La réception de (ping, Nbping) faire

Si m = Nbping alors	L1
Début <pong est perdu, il est régénéré>	L2
Nbping := Nbping + 1 ;	L3
Nbpong := - Nbping ;	L4
Fin	
Sinon m := nbping	L5
Fsi	

Fait

La réception de (pong, Nbpong) faire

<traitement analogue en intervertissant les rôles de ping et pong>

Fait

La rencontre des deux jetons faire

Nbping := Nbping + 1 ;	L6
Nbpong := Nbpong - 1 ;	L7

Fait

Corrigé :

Exercice 8

1. Quelles sont les propriétés comportementales des liaisons ?
2. Soit une liaison ne vérifiant pas une propriété H_i et soit un algorithme distribué nécessitant cette propriété pour son bon fonctionnement ; Quelle solution proposeriez vous pour l'utilisation de cet algorithme dans ces conditions ?

Corrigé :

Exercice 9

1. Donnez les avantages et les inconvénients des algorithmes à jeton résolvant le problème de la section critique dans un environnement distribué ?
2. Dans le même contexte, discutez les conditions pour lesquelles une topologie en anneau est avantageuse par rapport aux autres topologies ?

3. Quelles sont les topologies de réseaux que vous connaissez ? Quels sont les avantages et les inconvénients de chacune d'elles ?

Corrigé :

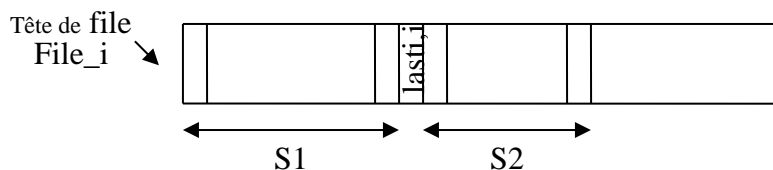
Exercice 10

Dans cet exercice on utilise l'algorithme de Carvalho et Roucairol pour gérer l'accès à la section critique de trois processus P1, P2 et P3 dont les horloges sont toutes égales à 0. On suppose que les ensembles R1, R2 et R3 sont comme suit : $R1 = \{\}$, $R2 = \{1\}$ et $R3 = \{1,2\}$.

1. Donnez la répartition des permissions sur les différents sites ?
2. Supposons que le processus P2 appelle la procédure acquérir et arrive à rentrer et sortir de la section critique alors que les états des processus P1 et P3 demeurent égales à « dehors ». Quelles sont les valeurs des horloges H1, H2 et H3 ainsi que la nouvelle répartition des permissions dans l'état résultant ?
3. A partir de l'état résultant, nous supposons que le processus P1 appelle la procédure acquérir et arrive à rentrer et sortir de sa section critique alors que les états des processus P2 et P3 demeurent égales à « dehors ». Quelles sont les valeurs des horloges H1, H2 et H3 ainsi que la nouvelle répartition des permissions dans l'état résultant ?
4. Que peut-on conclure ?

Corrigé :

Exercice 11: Soit l'algorithme mixte vu dans le cours. Soit la configuration suivante d'un processus P_i .



1. Dans le cas où l'état du processus P_i est *demandeur*. Que représentent les ensembles S1 et S2 ? Discutez l'état des permissions données par P_i ainsi que les valeurs des estampilles des éléments de S1 et de S2 vis-à-vis de celle de la requête de P_i ?
2. Même question dans le cas où l'état du processus P_i est *dedans* ?
3. Dans le cas où l'état du processus P_i est dehors. Dessinez la file *File_i* ?

Corrigé :

Exercice 12

Soit un système composé de N sites reliés par un anneau unidirectionnel. On suppose que les liaisons sont fiables. On se propose dans cet exercice d'écrire un algorithme résolvant le problème de la section critique. Ce dernier est basé sur l'existence d'un message particulier appelé **Jeton** qui est en mouvement circulaire perpétuel sur l'anneau.

1. Outre le message Jeton, quelles sont les variables (éventuellement une) utilisées par cet algorithme ?

2. Quelles sont les événements qui sont à l'initiative d'un processus ?
3. Quelles sont les événements externes qu'un processus peut subir ?
4. Pour chaque événement identifié précédemment, donnez la procédure à exécuter par le processus ?
5. L'exécution de chaque procédure doit remplir une propriété particulière, laquelle ?
6. Discuter les propriétés d'un tel algorithme en fonction de **N** et **T** (temps moyen de transit d'un message entre deux sites voisins).
7. Discutez les performances de l'algorithme en fonction de **N**.
8. Quels sont les problèmes engendrés par l'absence de la propriété de fiabilité des liaisons ? Sachant que nous n'avons pas le moyen pour rendre les liaisons fiables, quelle solution proposeriez-vous ? quels problèmes sous-jacents peuvent apparaître et comment remédier à ces problèmes ? Que peut-on conclure ?

Corrigé :

Exercice 13

Soit la donnée de **n** sites, à chaque site **i** est associé un processus **P_i** qui peut à tout moment invoqué l'exécution d'une opération **op_j** parmi **m** opérations **op₁, op₂, ..., op_m**. Le problème qui se pose est induit par la non compatibilité de certaines opérations. En d'autres termes deux opérations incompatibles invoquées chacune par un site doivent être exécutées en exclusion mutuelle.

Les relations d'exclusion entre ces opérations sont définies par une matrice **Compat**, dite de compatibilité, booléenne et symétrique tel que :

Compat(op_i, op_j) ⇔ op_i et op_j ne sont pas exclusives (elles sont compatibles)

Les messages de requêtes émis par un site **j** transportent, outre les estampilles, le nom de l'opération **op_k** qu'il veut exécuter. A la réception d'un tel message, un site **i** va retarder le renvoi de sa permission s'il a une opération en cours incompatible avec **op_k** et si son opération est dotée d'une estampille inférieure.

Le travail demandé consiste à adapter l'algorithme de Ricart et Agrawala pour la gestion de l'exécution de ces opérations par ces processus distribués. Pour cela chaque site **i** est doté, en plus des variables utilisées dans l'algorithme sus cité, de la variable locale **Last-op_i** qui mémorise le nom de la dernière opération invoquée par ce site (**op₁, op₂, ..., op_m**).

1. Rappelez les variables utilisées dans l'algorithme de Ricart et Agrawala en donnant le rôle de chacune d'elles.
2. Dans le cas de ce problème, quelle est la structure d'un message requête envoyé par un site **i** à un site **j**.
3. Donnez l'expression calculant la priorité d'un processus **i** recevant une requête de la part d'un processus **j**.
4. Donnez les algorithmes des énoncés régissant le comportement d'un site **i**, à savoir : Lors d'un appel à op_k, lors de la terminaison de op_k, lors de la réception d'une requête de la part d'un site j, lors de la réception d'une permission de la part d'un site j.

Corrigé :