



# Exclusion mutuelle

## Master 2 Informatique - UFR S.A.T

Pr. Ousmane THIARE

ousmane.thiare@ugb.edu.sn  
<http://www.ousmanethiare.com/>

16 avril 2020

# Chapitre 4 : Exclusion mutuelle

## Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

- 1 Rappel
- 2 Algorithme de la boulangerie
- 3 Exclusion mutuelle basée sur un jeton
- 4 Exclusion mutuelle par liste d'attente répartie



## Rappel

Algorithme de la  
boulangerie

Exclusion  
mutuelle basée  
sur un jeton

Exclusion  
mutuelle par  
liste d'attente  
répartie

# Chapitre 4 : Exclusion mutuelle

# Rappel

## Propriétés des protocoles d'exclusion mutuelle

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

**Définition** : à tout instant un processus au plus se trouve en section critique (SC).

**Atteignabilité** : si plusieurs processus sont bloqués en attente de la S.C. alors qu'aucun processus n'est en S.C, alors l'un d'eux doit y accéder en un temps fini.

**Progression** : un processus en attente accède à la section critique en un temps fini.

**Indépendance des parties conflictuelles et non conflictuelles** : un processus hors de la S.C ou du protocole d'entrée ne doit pas influencer sur le protocole d'exclusion mutuelle.

**Banalisation de la solution** : aucun processus ne joue le rôle privilégié.

Chaque fois que l'on proposera une solution, il faudra démontrer que ces propriétés sont bien respectées (pas toujours trivial !!).



# Rappel

## Cas des monoprocesseurs (ou mono-site)

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On utilise le plus souvent des solutions du type :

- variable de condition (attente active)

On celles-ci nécessitent des solution matérielles telles que soit une mémoire commune, soit un mécanisme de blocage du bus ou le masquage d'interruption  $\implies$  ceci est bien évidemment impossible sur une architecture faiblement couplée  $\implies$  utilisation de solution entièrement logicielle : protocole d'exclusion mutuelle.



# Rappel

## Cas des monoprocesseurs (ou mono-site)

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On utilise le plus souvent des solutions du type :

- variable de condition (attente active)
- sémaphore

On celles-ci nécessitent des solution matérielles telles que soit une mémoire commune, soit un mécanisme de blocage du bus ou le masquage d'interruption  $\implies$  ceci est bien évidemment impossible sur une architecture faiblement couplée  $\implies$  utilisation de solution entièrement logicielle : protocole d'exclusion mutuelle.



# Rappel

## Cas des monoprocesseurs (ou mono-site)

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On utilise le plus souvent des solutions du type :

- variable de condition (attente active)
- sémaphore
- moniteur

On celles-ci nécessitent des solution matérielles telles que soit une mémoire commune, soit un mécanisme de blocage du bus ou le masquage d'interruption  $\implies$  ceci est bien évidemment impossible sur une architecture faiblement couplée  $\implies$  utilisation de solution entièrement logicielle : protocole d'exclusion mutuelle.



# Rappel

## Cas des monoprocesseurs (ou mono-site)

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On utilise le plus souvent des solutions du type :

- variable de condition (attente active)
- sémaphore
- moniteur
- test & set (attente active)

On celles-ci nécessitent des solution matérielles telles que soit une mémoire commune, soit un mécanisme de blocage du bus ou le masquage d'interruption  $\implies$  ceci est bien évidemment impossible sur une architecture faiblement couplée  $\implies$  utilisation de solution entièrement logicielle : protocole d'exclusion mutuelle.





# Rappel

## Cas des monoprocesseurs (ou mono-site)

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Deux possibilités :

- protocole centralisé (le responsable de la ressource assure l'exclusion mutuelle). Problème : il faut qu'il y ait un tel responsable, qui en plus peut être un goulot d'étranglement



# Rappel

## Cas des monoprocesseurs (ou mono-site)

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Deux possibilités :

- protocole centralisé (le responsable de la ressource assure l'exclusion mutuelle). Problème : il faut qu'il y ait un tel responsable, qui en plus peut être un goulot d'étranglement
- protocole totalement décentralisé : on retrouvera ici, les deux grandes familles d'algorithmes par jeton ou estampille. Plus un algorithme particulier dans le cas où l'architecture permet le "partage" d'une mémoire (algorithme de la boulangerie).



# Algorithme de la boulangerie

## Principe

### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Lorsque l'on entre dans une boulangerie (ça marche aussi avec une boucherie...) deux cas :

- soit il y a un système de ticket (genre Casino)  $\implies$  ceci peut être assimilé à un serveur de ressources : pas intéressant

L'algorithme de la boulangerie est basé sur ce principe.



# Algorithme de la boulangerie

## Principe

### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Lorsque l'on entre dans une boulangerie (ça marche aussi avec une boucherie...) deux cas :

- soit il y a un système de ticket (genre Casino)  $\implies$  ceci peut être assimilé à un serveur de ressources : pas intéressant
- soit lorsqu'on entre, on s'affecte un "numéro" en "regardant" les numéros des autres, puis on attend son tour.

L'algorithme de la boulangerie est basé sur ce principe.



# Algorithme de la boulangerie

## Algorithme

### Rappel

### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Il nécessite l'utilisation de simulation d'une mémoire partagée accessible aux processus : les variables mises dans ces mémoires par un processus sont lisibles (non modifiables) par tous les autres.

Soit **num[1..N]** : un tableau de N entiers initialisés à 0  
Un processus  $P_i$  doit pouvoir "lire" le tableau mais ne peut en modifier que la case **num[i]**

Pour implémenter ce tableau, on remplace l'accès direct à **num[i]**,  $S_i$  peut donc modifier **num[i]** sans problème.

Lorsqu'un site  $S_k$  veut connaître la valeur de **num[i]**,  $i \neq k$ , il la demandera à  $S_j$ .

On note *Entree()* la fonction de demande de la S.C, *SC()* la fonction d'utilisation et *Sortie()* la fonction de libération de la SC.



# Algorithme de la boulangerie

## Algorithme

### Rappel

### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

## L'algorithme

▷ *Entree()*

{

//  $P_i$  est entrant :

choix[i]=vrai

$num[i] = 1 + \max(num[0], \dots, num[N - 1])$  // par lecture du tableau

choix[i]=faux

//  $P_i$  en attente :

for(j=1 ; j  $\neq$  i ; j=N-1){

{attendre que (choix[j]  $\neq$  vrai)

if (num[i,i]>(num[j],j) && (num[j]  $\neq$  0)

attendre que num[j] redevienne égal à 0

}



# Algorithme de la boulangerie

## Algorithme

Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

## L'algorithme

▷ *Sortie()* {  $num[i] = 0$  }



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### pas d'interblocage

Supposons que  $P_i$  et  $P_k$  soient en attente et que  $P_i$  était en attente avant  $P_k$ .

Alors  $\text{num}[i] < \text{num}[k] \implies P_i$  s'exécutera avant  $P_k \rightarrow$  le protocole se comporte comme une FIFO  $\rightarrow$  un processus ne peut qu'avancer dans la liste d'attente  $\rightarrow$  il s'exécutera dans un temps fini.

De plus, même si  $P_i$  et  $P_k$  ont exécuté au même moment leur partie entrante  $\rightarrow$  ils peuvent avoir le même numéro  $\rightarrow$  pas grave : le choix se fera sur le numéro de processus qui lui est différent pour tous les processus.





### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

Montrons que : si  $P_i$  est en SC et  $P_k$  est en attente, alors  $P_k$  ne peut entrer en SC.

Cela revient à montrer que nécessairement :

$(\text{num}[k],k) > (\text{num}[i],i)$  et  $\text{num}[i] \neq 0$  lorsque  $P_k$  le testera.

Soit :

- $T_{a0}$  : l'instant où  $P_i$  écrit  $\text{num}[i]$



### Rappel

#### Algorithme de la boulangerie

#### Exclusion mutuelle basée sur un jeton

#### Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

Montrons que : si  $P_i$  est en SC et  $P_k$  est en attente, alors  $P_k$  ne peut entrer en SC.

Cela revient à montrer que nécessairement :

$(\text{num}[k],k) > (\text{num}[i],i)$  et  $\text{num}[i] \neq 0$  lorsque  $P_k$  le testera.

Soit :

- $T_{a0}$  : l'instant où  $P_i$  écrit  $\text{num}[i]$
- $T_{a1}$  : l'instant où  $P_i$  lit son choix[k] pour la dernière fois (pour  $j=k$ )  $\implies$  à ce moment, choix[k]=faux



### Rappel

#### Algorithme de la boulangerie

#### Exclusion mutuelle basée sur un jeton

#### Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

Montrons que : si  $P_i$  est en SC et  $P_k$  est en attente, alors  $P_k$  ne peut entrer en SC.

Cela revient à montrer que nécessairement :

$(\text{num}[k],k) > (\text{num}[i],i)$  et  $\text{num}[i] \neq 0$  lorsque  $P_k$  le testera.

Soit :

- $T_{a0}$  : l'instant où  $P_i$  écrit  $\text{num}[i]$
- $T_{a1}$  : l'instant où  $P_i$  lit son choix[k] pour la dernière fois (pour  $j=k$ )  $\implies$  à ce moment, choix[k]=faux
- $T_{a2}$  : l'instant où  $P_i$  termine sa dernière exécution pour  $j=k$  de la seconde attente : attendre que  $\text{num}[j]$  redevienne égal à 0



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

remarque 1 :  $T_{a1} < T_{a2}$  :

- $T_{k1}$  : l'instant où  $P_k$  entre dans la zone "entrant"



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

remarque 1 :  $T_{a1} < T_{a2}$  :

- $T_{k1}$  : l'instant où  $P_k$  entre dans la zone "entrant"
- $T_{k2}$  : l'instant où  $P_k$  écrit num[k]



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

remarque 1 :  $T_{a1} < T_{a2}$  :

- $T_{k1}$  : l'instant où  $P_k$  entre dans la zone "entrant"
- $T_{k2}$  : l'instant où  $P_k$  écrit num[k]
- $T_{k3}$  : l'instant où  $P_k$  sort de la zone "entrant"



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

remarque 1 :  $T_{a1} < T_{a2}$  :

- $T_{k1}$  : l'instant où  $P_k$  entre dans la zone "entrant"
- $T_{k2}$  : l'instant où  $P_k$  écrit num[k]
- $T_{k3}$  : l'instant où  $P_k$  sort de la zone "entrant"
- $T_{k4}$  : l'instant où  $P_k$  teste  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$



# Algorithme de la boulangerie

## Preuve

### Rappel

### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

**remarque 2 :**  $T_{k1} < T_{k2} < T_{k3} < T_{k4}$

**remarque 3 :** après  $T_{a0}$ , num[i] ne change plus. De même, après  $T_{k2}$ , num[k] ne change plus.

**remarque 4 :** à  $T_{a1}$  on a obligatoirement choix[k]=faux, d'où  $T_{a1}$  : ne peut pas être compris entre  $T_{k1}$  et  $T_{k3}$   
d'où

- soit  $T_{a1} < T_{k1}$  ( $P_k$  est entré en SC avant même que  $P_k$  soit entrant) et alors num[i]<num[k] **CQFD**





### Rappel

### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### exclusion mutuelle

**remarque 2 :**  $T_{k1} < T_{k2} < T_{k3} < T_{k4}$

**remarque 3 :** après  $T_{a0}$ ,  $\text{num}[i]$  ne change plus. De même, après  $T_{k2}$ ,  $\text{num}[k]$  ne change plus.

**remarque 4 :** à  $T_{a1}$  on a obligatoirement  $\text{choix}[k]=\text{faux}$ , d'où  $T_{a1}$  : ne peut pas être compris entre  $T_{k1}$  et  $T_{k3}$   
d'où

- soit  $T_{a1} < T_{k1}$  ( $P_k$  est entré en SC avant même que  $P_k$  soit entrant) et alors  $\text{num}[i] < \text{num}[k]$  **CQFD**
- soit  $T_{k3} < T_{a1}$  ( $P_i$  a fini son calcul de  $\text{num}[k]$  avant que  $P_i$  n'entre en SC)



# Algorithme de la boulangerie

## Preuve

### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ . D'où

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq 0$ .



# Algorithme de la boulangerie

## Preuve

### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ .  
D'où

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .



### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ .  
D'où

■  $T_{a0} < T_{k4}$

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .



### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ . D'où

- $T_{a0} < T_{k4}$
- $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{k2}$

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .



# Algorithme de la boulangerie

## Preuve

### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ . D'où

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq -1$ . D'où on l'aura obligatoirement en  $T_{k4}$  **CQFD**.



# Algorithme de la boulangerie

## Preuve

### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ . D'où

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq -1$ . D'où on l'aura obligatoirement en  $T_{k4}$  **CQFD**.



### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ . D'où

■  $T_{a0} < T_{k4}$

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i], i) > (\text{num}[k], k)$  et  $\text{num}[k] \neq -1$ . D'où on l'aura obligatoirement en  $T_{k4}$  **CQFD**.





### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

## exclusion mutuelle

et alors  $T_{k2} < T_{k3} < T_{a1} < T_{a2} \implies T_{k2} < T_{a2}$  donc à  $T_{a1}$ ,  $P_i$  a lu pour la dernière fois le  $\text{num}[k]$  qui a été choisi à  $T_{k2}$  d'où, comme il a passé cette instruction en  $T_{a2}$  d'où, comme il a passé cette instruction en  $T_{a2}$ , il a OBLIGATOIREMENT trouvé  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq 0$ .

Comme  $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{a0}$ , on a nécessairement  $T_{a0} < T_{k2}$ . D'où

- $T_{a0} < T_{k4}$
- $\text{num}[k]$  n'a pas changé depuis  $T_{k2}$  et  $\text{num}[i]$  n'a pas changé depuis  $T_{k2}$

d'où en  $T_{k2}$  on avait déjà  $(\text{num}[i],i) > (\text{num}[k],k)$  et  $\text{num}[k] \neq -1$ . D'où on l'aura obligatoirement en  $T_{k4}$  **CQFD**.



Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

## Inconvénient de cet algorithme

- `num[i]` peut devenir très grand



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

## Inconvénient de cet algorithme

- `num[i]` peut devenir très grand
- mise en place de la mémoire commune



### Rappel

#### Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### Inconvénient de cet algorithme

- `num[i]` peut devenir très grand
- mise en place de la mémoire commune
- attente active : le site en attente doit scruter régulièrement les tableaux `num` et `choix`



# Exclusion mutuelle basée sur un jeton

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Un jeton unique circule dans le réseau. Seul le processus possédant celui-ci est autorisé à entrer en SC.

- un processus qui veut entrer en SC : diffuse une requête et attend  $\rightarrow$  dès qu'il reçoit le jeton, il entre en SC
- un processus qui reçoit une requête :
  - mémorise cette requête puis
  - s'il a le jeton et si il est en SC : terminé, rien à faire



# Exclusion mutuelle basée sur un jeton

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Un jeton unique circule dans le réseau. Seul le processus possédant celui-ci est autorisé à entrer en SC.

- un processus qui veut entrer en SC : diffuse une requête et attend  $\rightarrow$  dès qu'il reçoit le jeton, il entre en SC
- un processus qui reçoit une requête :
  - s'il a le jeton et si il est en SC : terminé, rien à faire
  - s'il a le jeton et s'il n'est pas en SC : il ré-exécute *Sortie()*



# Exclusion mutuelle basée sur un jeton

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Un jeton unique circule dans le réseau. Seul le processus possédant celui-ci est autorisé à entrer en SC.

- un processus qui veut entrer en SC : diffuse une requête et attend  $\rightarrow$  dès qu'il reçoit le jeton, il entre en SC
- un processus qui reçoit une requête :
  - s'il a le jeton et si il est en SC : terminé, rien à faire
  - s'il a le jeton et s'il n'est pas en SC : il ré-exécute *Sortie()*
  - s'il n'a pas le jeton : terminé, rien à faire



# Exclusion mutuelle basée sur un jeton

## Principe

### Rappel

Algorithme de la boulangerie

**Exclusion mutuelle basée sur un jeton**

Exclusion mutuelle par liste d'attente répartie

- un processus qui quitte la SC :
- si il a mémorisé une requête, il envoie le jeton





# Exclusion mutuelle basée sur un jeton

## Principe

### Rappel

Algorithme de la  
boulangerie

**Exclusion  
mutuelle basée  
sur un jeton**

Exclusion  
mutuelle par  
liste d'attente  
répartie

- un processus qui quitte la SC :
  - si il a mémorisé une requête, il envoie le jeton
  - sinon : terminé, rien à faire



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On définit les types :

- JETON : un tableau de  $N$  d'entiers (pour une variable  $J$  de ce type,  $(J[i], i)$  correspondra à une date sur le site  $S_i$ ).

Chaque processus  $P_i$  dispose :



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On définit les types :

- JETON : un tableau de  $N$  d'entiers (pour une variable  $J$  de ce type,  $(J[i], i)$  correspondra à une date sur le site  $S_i$ ).
- REQUETE = {entier date ; entier émetteur}

Chaque processus  $P_i$  dispose :



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On définit les types :

- JETON : un tableau de N d'entiers (pour une variable J de ce type,  $(J[i], i)$  correspondra à une date sur le site  $S_i$ ).
- REQUETE={entier date ; entier émetteur}

Chaque processus  $P_i$  dispose :

- de deux variables du type JETON : tampon et demande



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On définit les types :

- JETON : un tableau de N d'entiers (pour une variable J de ce type,  $(J[i], i)$  correspondra à une date sur le site  $S_i$ ).
- REQUETE={entier date ; entier émetteur}

Chaque processus  $P_i$  dispose :

- de deux variables du type JETON : tampon et demande
- deux variables booléennes : dedans et jeton\_present initialisé à  $(mon\_numero==1)$  **(4)**



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

On définit les types :

- JETON : un tableau de N d'entiers (pour une variable J de ce type,  $(J[i], i)$  correspondra à une date sur le site  $S_i$ ).
- REQUETE={entier date ; entier émetteur}

Chaque processus  $P_i$  dispose :

- de deux variables du type JETON : tampon et demande
- deux variables booléennes : dedans et jeton\_present initialisé à  $(mon\_numero==1)$  **(4)**
- une variable entière : estampille



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

```
L'algorithme pour un  $P_i$  est le suivant :  $\triangleright$  Entree()  
{ REQUETE requete ;  
  estampille++  
  requete.emetteur = mon_numero; requete.date =  
  estampille ;  
  demande[mon_numero] = estampille ;  
  diffuser(requete) /* uniquement aux autres sites */ ;  
  Si(jeton_present == faux) attendre(tampon) ;  
  dedans=vrai ;  
  jeton_present = vrai ;  
}
```



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

```
▷ Sortie()
{ tampon[mon_numero] = estampille;    (2)
  dedans=faux ;
  pour  $j = \textit{mon\_numero} + 1, ..n, 1, \dots, \textit{mon\_numero} - 1$     (3)
  { Si (demande[j] > tampon[j]) et (jeton_present ==
    vrai)    (1)
    { jeton_present=faux ; envoyer(tampon,j) ;}
  }
}
```





# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la  
boulangerie

Exclusion  
mutuelle basée  
sur un jeton

Exclusion  
mutuelle par  
liste d'attente  
répartie

▷ Lorsqu'un site  $P_i$  reçoit un message il exécute :

```
Traiter Requete(REQUETTE requete)
{ entier k=requete.emetteur ;
  demande[k]=max(demande[k], requete.date) ;
  Si(jeton_present == vrai)et(dedans == faux){Sortie()}}
```



# Exclusion mutuelle basée sur un jeton

## Algorithme

### Rappel

Algorithme de la  
boulangerie

Exclusion  
mutuelle basée  
sur un jeton

Exclusion  
mutuelle par  
liste d'attente  
répartie

▷ Lorsqu'un site  $P_i$  reçoit un message il exécute :

```
Traiter Requete(REQUETTE requete)
{ entier k=requete.emetteur ;
  demande[k]=max(demande[k], requete.date) ;
  Si(jeton_present == vrai)et(dedans == faux){Sortie()}}
```



# Exclusion mutuelle basée sur un jeton

## Algorithme

Avec :

- **(1)** : Chaque site mémorise la date de la dernière demande des autres sites dans `demande[j]`.  
Dans le processus possédant le jeton, le tableau tampon contient les dates de la dernière possession du jeton par les autres processus donc si `demande[i] < tampon[i]`, le processus  $P_i$  a été satisfait de sa demande, il n'y a pas lieu de lui envoyer le jeton

Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie



# Exclusion mutuelle basée sur un jeton

## Algorithme

Avec :

- **(1)** : Chaque site mémorise la date de la dernière demande des autres sites dans `demande[j]`.  
Dans le processus possédant le jeton, le tableau tampon contient les dates de la dernière possession du jeton par les autres processus donc si `demande[i] < tampon[i]`, le processus  $P_i$  a été satisfait de sa demande, il n'y a pas lieu de lui envoyer le jeton
- **(2)** : Mise à jour du tampon qui sera envoyé comme jeton : le processus  $y$  met la date locale de sa dernière possession du jeton

Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie



# Exclusion mutuelle basée sur un jeton

## Algorithme

Avec :

- **(1)** : Chaque site mémorise la date de la dernière demande des autres sites dans `demande[j]`.  
Dans le processus possédant le jeton, le tableau tampon contient les dates de la dernière possession du jeton par les autres processus donc si `demande[i] < tampon[i]`, le processus  $P_i$  a été satisfait de sa demande, il n'y a pas lieu de lui envoyer le jeton
- **(2)** : Mise à jour du tampon qui sera envoyé comme jeton : le processus  $y$  met la date locale de sa dernière possession du jeton
- **(3)** : Permet de "balayer" les sites de façon circulaire : attente bornée

Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie



# Exclusion mutuelle basée sur un jeton

## Algorithme

Avec :

- **(1)** : Chaque site mémorise la date de la dernière demande des autres sites dans `demande[j]`.  
Dans le processus possédant le jeton, le tableau tampon contient les dates de la dernière possession du jeton par les autres processus donc si `demande[i] < tampon[i]`, le processus  $P_i$  a été satisfait de sa demande, il n'y a pas lieu de lui envoyer le jeton
- **(2)** : Mise à jour du tampon qui sera envoyé comme jeton : le processus  $y$  met la date locale de sa dernière possession du jeton
- **(3)** : Permet de "balayer" les sites de façon circulaire : attente bornée
- **(4)** : initialisation : seul le processus  $P_1$  dispose initialement du jeton.

Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie



# Exclusion mutuelle basée sur un jeton

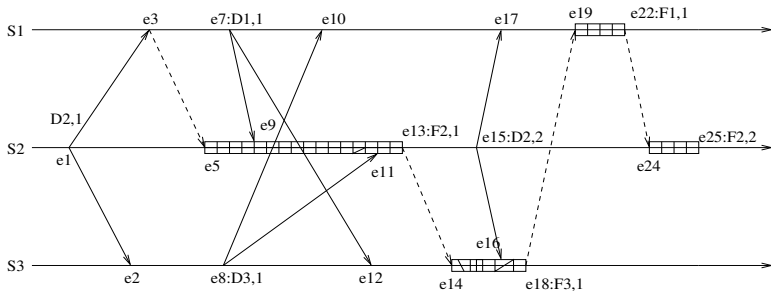
## Exemple

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie



# Exclusion mutuelle basée sur un jeton

## Exemple

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Avec :

	0	1	2	3	5	7	8	9	10	11	12
dedans	F			F		F			F		
estampille	0			0		1			1		
jeton	V			F		F			F		
demande	0,0,0			0,1,0		1,1,0			1,1,1		
tampon	0,0,0			0,0,0		0,0,0			0,0,0		
Action				T(0,0,0)→ S2		R(1,1)					
dedans	F	F			V			V		V	
estampille	0	1			1			1		1	
jeton	F	F			V			V		V	
demande	0,0,0	0,1,0			0,1,0			1,1,0		1,1,1	
tampon	0,0,0	0,0,0			0,0,0			0,0,0		0,0,0	
Action		R(1,2)			SC()						
dedans	F		F				F				F
estampille	0		0				1				1
jeton	F		F				F				F
demande	0,0,0		0,1,0				0,1,1				1,1,1
tampon	0,0,0		0,0,0				0,0,0				0,0,0
Action							R(1,3)				





# Exclusion mutuelle basée sur un jeton

## Exemple

### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

	13	14	15	16	17	18	19	22	24	25
dedans					F		V	F		
estampille					1		1	1		
jeton					F		V	F		
demande					1,2,1		1,2,1	1,2,1		
tampon					0,0,0		1,1,1	1,1,1		
Action							SC()	T(1,1,1)→S2		
dedans	F		F						V	F
estampille	1		2							
jeton	F		F						V	V
demande	1,1,1		1,2,1						1,2,1	1,2,1
tampon	0,1,0		0,1,0						1,1,1	1,2,1
Action	T(0,1,0)→S3		R(2,2)						SC()	
dedans		V		V		F				
estampille		1		1		1				
jeton		V		V		F				
demande		1,1,1		1,2,1		1,2,1				
tampon		0,1,0		0,1,0		0,1,1				
Action		SC()				T(0,1,1)→S1				



# Exclusion mutuelle basée sur un jeton

## Preuve

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

**Exclusion mutuelle assurée ?** A tout instant, il y a au plus une des variables locales `jeton_present` qui vaut vrai. En effet, au départ (4) seul `jeton_present` de  $P_1$  est vrai et ensuite seule la réception du tampon (suite à `attendre_jeton`) modifie cette variable. Comme cet envoi est nécessairement précédé d'un `jeton_present=faux`, la propriété est vraie.

**Progression ?** S'il n'y a pas de processus dans la SC, alors un processus en attente recevra-t-il le jeton ? Evident : dès qu'un processus a terminé, il transmet le jeton sur simple demande ou après mémorisation.



# Exclusion mutuelle basée sur un jeton

## Preuve

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

**Attente bornée ?** S'il y a plusieurs processus en attente, l'ordre d'"arrivée" est-il respecté pour l'entrée en SC ?  
A partir du moment où tous les messages de requêtes sont arrivés à destination (temps fini), la valeur de la  $j$ -ième entrée des tableaux demande est supérieure à celle de la  $j$ -ième entrée du jeton. Comme le processus qui tient le jeton le transmet en l'explorant de façon circulaire, un processus ne peut être précédé que de  $(N-1)$  processus (dans l'ordre d'entrée dans la SC).

**Remarques :** Problème de la perte du jeton.

Coût en message : 0 ou  $N$  ( $N-1$  pour le requête, 1 pour le jeton) par demande.



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### Réseau FIFO

Chaque site gère une copie de la file d'attente (ou une "vision" de celle-ci) mise à jour par des messages de requêtes et de libération de la SC.

Chaque site doit :

- recevoir tous ces messages de requêtes et de libérations de tous les autres sites

⇒ utilisation d'horloges logiques



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### Réseau FIFO

Chaque site gère une copie de la file d'attente (ou une "vision" de celle-ci) mise à jour par des messages de requêtes et de libération de la SC.

Chaque site doit :

- recevoir tous ces messages de requêtes et de libérations de tous les autres sites
- savoir les mettre *dans le bon ordre*

⇒ utilisation d'horloges logiques



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Chaque site  $S_i$  gère un tableau  $F_i[1..N]$  à  $N$  entrées où chaque  $F_i[j]$  (y compris  $F_i[i]$ ) contient un message en provenance de  $S_j$ .

Les messages sont sous la forme : *(type du message, date du message, site)* où le type d'un message peut être : *hors\_SC, requete\_SC* ou *Ack*.

à  $T_0 = 0$ ,  $F_i[j] = (\text{hors\_SC}, 0, j)$  pour tout  $i$  et tout  $j$ .



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

- ▷ **Demande d'entrée en section critique** par un site  $S_i$ .
  - $S_i$  émet en diffusion le message (requete\_SC,  $H_i$ ,  $i$ ) vers les autres sites
  
- ▷ **Sortie de la section critique** par un site  $S_i$ .



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

- ▷ **Demande d'entrée en section critique** par un site  $S_i$ .
  - $S_i$  émet en diffusion le message  $(requete\_SC, H_i, i)$  vers les autres sites
  - ce message  $(requete\_SC, H_i, i)$  est mis en tête dans la file d'attente  $F_i[j]$ .
- ▷ **Sortie de la section critique** par un site  $S_i$ .





# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

- ▷ **Demande d'entrée en section critique** par un site  $S_i$ .
- $S_i$  émet en diffusion le message (requete\_SC,  $H_i, i$ ) vers les autres sites
- ce message (requete\_SC,  $H_i, i$ ) est mis en tête dans la file d'attente  $F_i[j]$ .
- $H_i + +$
- ▷ **Sortie de la section critique** par un site  $S_i$ .



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

- ▷ **Demande d'entrée en section critique** par un site  $S_i$ .
  - $S_i$  émet en diffusion le message (requete\_SC,  $H_i, i$ ) vers les autres sites
  - ce message (requete\_SC,  $H_i, i$ ) est mis en tête dans la file d'attente  $F_i[j]$ .
  - $H_i + +$
- ▷ **Sortie de la section critique** par un site  $S_i$ .
  - $S_i$  émet (hors\_SC,  $H_i, i$ ) vers les autres sites



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

▷ **Demande d'entrée en section critique** par un site  $S_i$ .

■  $S_i$  émet en diffusion le message  $(requete\_SC, H_i, i)$  vers les autres sites

■ ce message  $(requete\_SC, H_i, i)$  est mis en tête dans la file d'attente  $F_i[j]$ .

■  $H_i + +$

▷ **Sortie de la section critique** par un site  $S_i$ .

■  $S_i$  émet  $(hors\_SC, H_i, i)$  vers les autres sites

■  $H_i + +$



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

▷ **Réception d'un message** par un site  $S_j$ .

### Version 1

Sur un site  $S_i$ , à chaque événement de réception d'un message, l'horloge  $H_i$  est recalée puis

- si le message est du type  $(requete, H_j, j)$  : émission de  $(Ack, H_i, i)$  vers  $S_j$



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

▷ **Réception d'un message** par un site  $S_j$ .

### Version 1

Sur un site  $S_i$ , à chaque événement de réception d'un message, l'horloge  $H_i$  est recalée puis

- si le message est du type  $(requete, H_j, j)$  : émission de  $(Ack, H_i, i)$  vers  $S_j$
- si le message est du type  $(Ack, H_j, j)$



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

▷ **Réception d'un message** par un site  $S_j$ .

### Version 1

Sur un site  $S_i$ , à chaque événement de réception d'un message, l'horloge  $H_i$  est recalée puis

- si le message est du type (requete,  $H_j, j$ ) : émission de (Ack,  $H_i, i$ ) vers  $S_j$
- si le message est du type (Ack,  $H_j, j$ )
  - le message est placé en tête de  $F_i[j]$  si tête de  $F_i[j]$  n'est pas un message du type requete\_SC



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

▷ **Réception d'un message** par un site  $S_j$ .

### Version 1

Sur un site  $S_i$ , à chaque événement de réception d'un message, l'horloge  $H_i$  est recalée puis

- si le message est du type  $(requete, H_j, j)$  : émission de  $(Ack, H_i, i)$  vers  $S_j$
- si le message est du type  $(Ack, H_j, j)$ 
  - le message est placé en tête de  $F_i[j]$  si tête de  $F_i[j]$  n'est pas un message du type `requete_SC`
  - le message est ignoré sinon



# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### Version 2

Sur un site  $S_i$ , à chaque événement de réception d'un message, l'horloge  $H_i$  est recalée puis

- si le message est du type  $(requete, H_j, j)$  : émission de  $(Ack, H_i, i)$  vers  $S_j$  si tête de  $F_i[i]$  n'est pas un message du type `requete_SC`

Dans les deux cas un site  $S_i$  s'octroie le droit d'entrer en SC lorsque le message en tête  $F_i[i]$  est du type `requete_SC` et que son estampille est la plus ancienne de tous les messages en tête de  $F_i[j]$ ,  $i \neq j$ .





# Exclusion mutuelle par liste d'attente répartie

## Principe

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

### Version 2

Sur un site  $S_i$ , à chaque événement de réception d'un message, l'horloge  $H_i$  est recalée puis

- si le message est du type  $(requete, H_j, j)$  : émission de  $(Ack, H_i, i)$  vers  $S_j$  si tête de  $F_i[i]$  n'est pas un message du type `requete_SC`
- si le message est du type  $(Ack, H_j, j)$ , le message est placé en tête de  $F_i[j]$

Dans les deux cas un site  $S_i$  s'octroie le droit d'entrer en SC lorsque le message en tête  $F_i[i]$  est du type `requete_SC` et que son estampille est la plus ancienne de tous les messages en tête de  $F_i[j]$ ,  $i \neq j$ .



# Exclusion mutuelle par liste d'attente répartie

## Algorithme

### Rappel

#### Algorithme de la boulangerie

#### Exclusion mutuelle basée sur un jeton

#### Exclusion mutuelle par liste d'attente répartie

Pour un site  $S_i$

▷  $Init\_Site()$  {pour  $j = 1$  à  $N$  {  $Fi[j] = (Hors\_SC, 0, j)$  }

▷  $Entree\_SC()$

{  $Diffuser(requete\_SC, hi, i)$ ;  $Fi[i] =$   
 $(requete\_SC, hi, i)$ ;  $Hi ++$

$Attendre\_que (\forall j \neq i, estampille\ de\ Fi[i] < estampille\ Fi[j])$  }

▷  $Sortie\_SC()$  {  $Diffuser(Hors\_SC, Hi, i)$ ;  $Fi[i] =$   
 $(Hors\_SC, Hi, i)$ ;  $Hi ++$  }



# Exclusion mutuelle par liste d'attente répartie

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

```
Lorsqu'un site  $S_i$  reçoit un message il exécute
▷ Reception_message((type,h,j)) /*Algorithme version un*/
{if( $h > H_i$ ) $H_i = h$ ;  $H_i ++$ ;
switch (type) in
    {caserequete_SC :  $Fi[j] = requete\_SC, h, j$ )
    case Hors_SC :  $Fi[j]=Hors\_SC,h,j$ )
    case Ack :
Si( $type(Fi[j] \neq Requete\_SC)$ ) $Fi[j] = (Ack, h, j)$ 
}
}
```



# Exclusion mutuelle par liste d'attente répartie

## Algorithme

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

```
Pour un site  $S_i$ 
▷ Reception_message((type,h,j)) /* Algorithme version
deux */
{ if (h>Hi) Hi=h ; Hi++ ;
switch (type) in
{ case requete_SC :
  Fi[j]=(requete_SC,h,j)
  Si  $S_i$  n'est pas en SC, envoie (Ack,Hi,i) à  $S_j$ 
  case Hors_SC : Fi[j]=Hors_SC,h,j)
  case Ack : Si Fi[j]=(Ack,h,j)
}
}
```



# Exclusion mutuelle par liste d'attente répartie

## Algorithme

### Rappel

#### Algorithme de la boulangerie

#### Exclusion mutuelle basée sur un jeton

#### Exclusion mutuelle par liste d'attente répartie

Les ordres d'entrées en SC seront traités selon l'ordre total grâce à l'estampillage car lorsqu'un site  $S_i$  décide d'entrer en SC il ne peut exister "dans le système (réseau compris)" aucun message du type `requete_SC` de  $S_j$  émis avant le sien sinon  $S_i$  l'aurait nécessairement reçu avant l'acquittement de  $S_j$  car le réseau est FIFO.

De plus, la relation d'ordre étant acyclique, le protocole est exempt d'interblocage  $\implies$  atteignabilité.



# Exclusion mutuelle par liste d'attente répartie

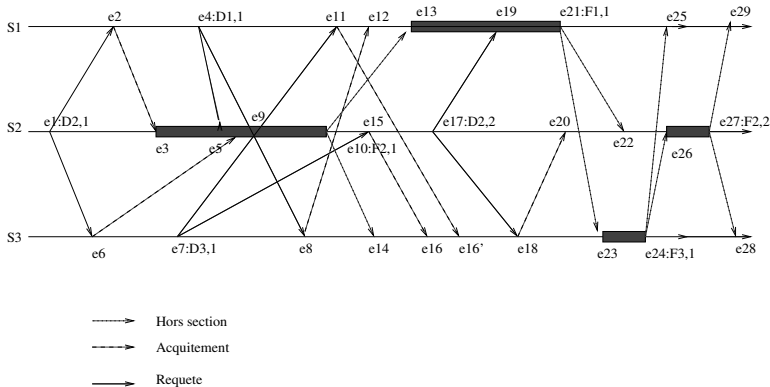
## Exemple

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie



# Exclusion mutuelle par liste d'attente répartie

## Exemple

### Rappel

Algorithme de la boulangerie

Exclusion mutuelle basée sur un jeton

Exclusion mutuelle par liste d'attente répartie

Nous donne

E	0	1	2	3	4	5	6	7	8
H1	0		1		2				
F1[1]	hs,0,1		hs,0,1		rq,1,1				
F2[2]	hs,0,2		rq,0,2		rq,0,2				
F3[3]	hs,0,3		hs,0,3		hs,0,3				
Action			E(ack,1,1)		E(rq,1,1), Att				
H2	0	1		2		3			
F2[1]	hs,0,1	hs,0,1		ack,1,1		rq,1,1			
F2[2]	hs,0,2			rq,0,2		rq,0,2			
F2[3]	hs,0,3	hs,0,3		hs,0,3		hs,0,3			
Action		E(rq,0,2), Att		SC( )					
H3	0						1	2	3
F3[1]	hs,0,1						hs,0,1	hs,0,1	rq,1,1
F3[2]	hs,0,2						rq,0,2	rq,0,2	rq,0,2
F3[3]	hs,0,3						hs,0,3	rq,1,3	rq,1,3
Action							E(ack,1,3)	E(rq,1,3), Att	E(ack,3,3)



# Exclusion mutuelle par liste d'attente répartie

## Exemple

### Rappel

### Algorithme de la boulangerie

### Exclusion mutuelle basée sur un jeton

### Exclusion mutuelle par liste d'attente répartie

E	9	10	11	12	13	14	15	16	16'	17	18
H1			3	4	5						
F1[1]			rq,1,1	rq,1,1	rq,1,1						
F1[2]			rq,0,2	rq,0,2	rq,0,2						
F1[3]			rq,1,3	rq,1,3	rq,1,3						
action			E(ack,3,1)		SC()						
H2	4	5					6			7	
F2[1]	rq,1,1	rq,1,1					rq,1,1			rq,1,1	
F2[2]	rq,0,2	hs,4,2					hs,4,2			rq,6,2	
F2[3]	ack,1,3	ack,1,3					rq,1,3			rq,1,3	
action		E(hs,4,2)					E(ack,6,2)			E(rq,6,2), Att	
H3						5		7	8		9
F3[1]						rq,1,1		rq,1,1	rq,1,1		rq,1,1
F3[2]						hs,4,2		ack,6,2	ack,5,2		rq,6,2
F3[2]						rq,1,3		rq,1,3	rq,1,3		rq,1,3
action											E(ack,9,3)

