Exclusion mutuelle

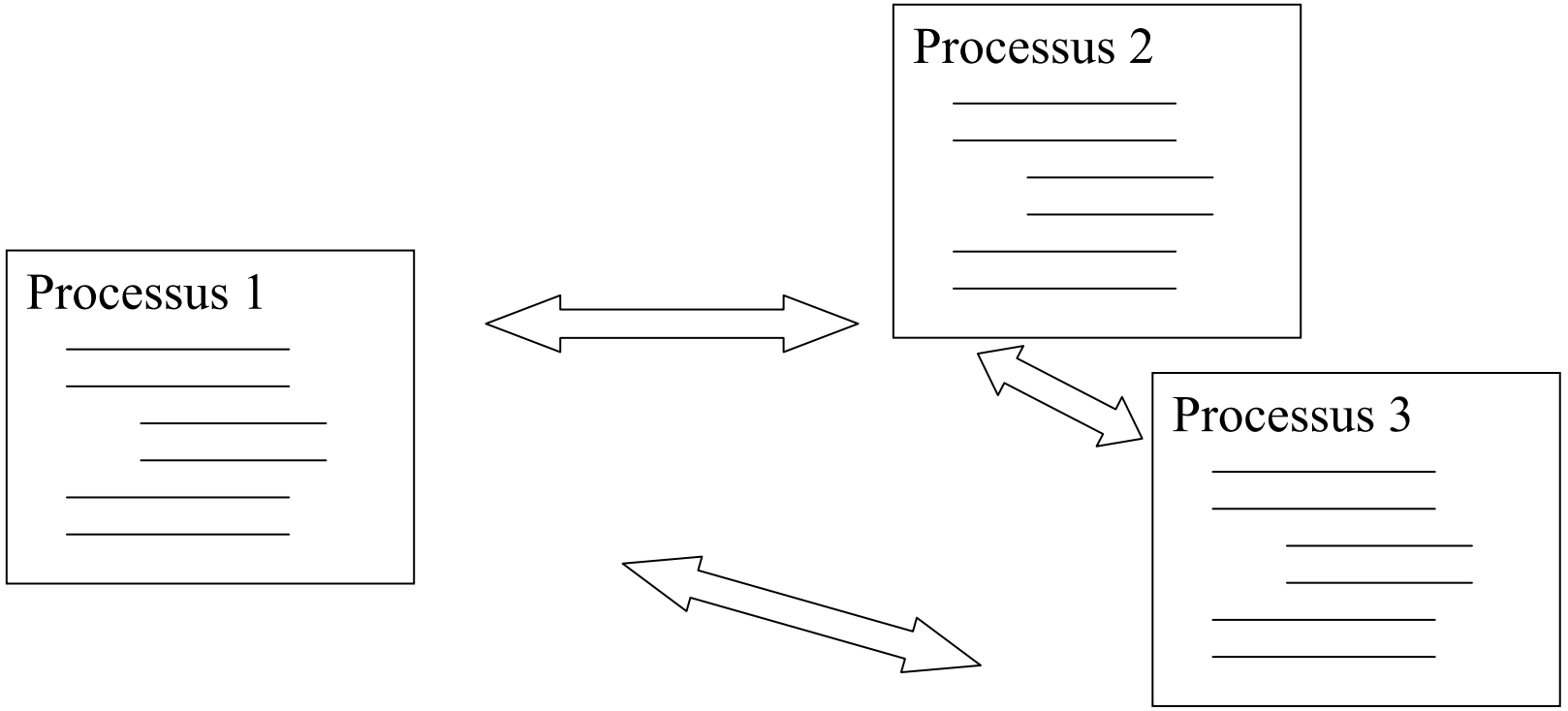
- ✓ Contexte de plusieurs processus s'exécutant en parallèle
- ✓ Accès à une ressource partagée par un seul processus à la fois

Exclusion mutuelle en distribué

- ✓ Accès à une ressource partagée distante par un seul processus à la fois
- ✓ Processus distribués
  - ➢ Requêtes et gestion d'accès via des messages échangés entre les processus
  - ➢ Nécessité de mettre en œuvre des algorithmes gérant ces échanges de messages pour assurer l'exclusion mutuelle

♦ *L'algorithmique distribuée :*

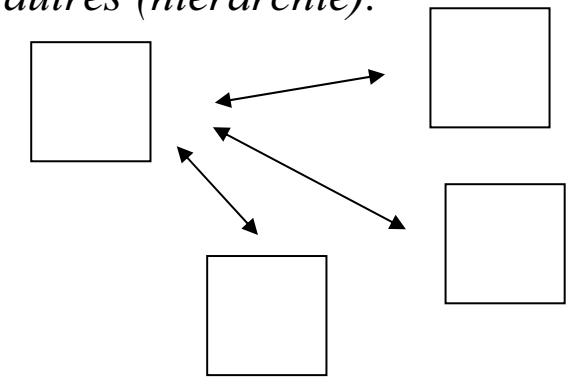
- ***Eléments de base :*** *processus ou sites et voies de communication*



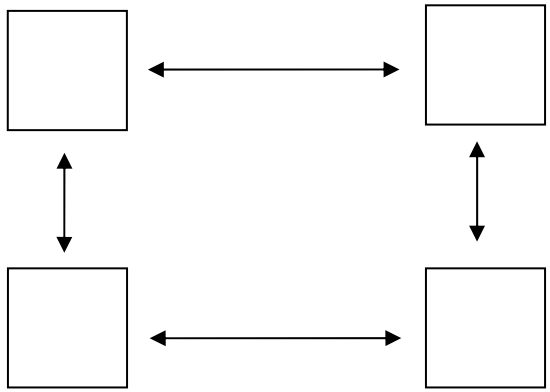
♦ Voies de communication (ou liaison, ou canaux)

- *Structures :*

- ***Etoile :*** *un processus particulier peut communiquer avec tous les autres (hiérarchie).*



- ***Anneau :*** *chaque processus communique avec ces deux voisins .*



- ***Arbre :*** *chaque processus communique avec ses fils et son père*
- ***Maillage complet :*** *chaque processus communique avec tous les processus*

Attention : Ne pas confondre la structure du réseau local avec la structure de communication de l'algorithme distribué.

♦ *Voies de communication :*

- *Propriétés comportementales : Hypothèses de simplification classiques*

- ***H1*** *: La transmission sur la voie se fait sans duplication de message.*

- ***H2*** *: La transmission se fait sans altération des messages.*

- ***H3*** *: Pour tout couple de processus communicants, l'ordre de réception des messages est identique à leur ordre d'émission → pas de déséquencement.*

- ***H4*** *: Le délai d'acheminement des messages est fini (bien qu'aléatoire) → pas de perte.*

- ***H5*** *: Le délai d'acheminement est borné.*

Positionnement du problème

♦ *l'objectif :*

- *Accorder un privilège à un processus parmis n (accès à une ressource non partageable…).*

♦ *Problèmes spécifiques :*

- Equité *(vivacité) : Tout processus ayant besoin de la ressource y aura accès au bout d'un temps fini → absence de blocage.*
- Sûreté *: A tout instant il y a au plus un processus accédant à la ressource.*
- Minimiser les échanges.

♦ *Problème d'équitabilité*

- *Deux classes d'algorithmes*

  - Algorithmes fondés sur les permissions

    Permissions individuelles
    Permission d'arbitre
    + : rapidité
    - : compléxité

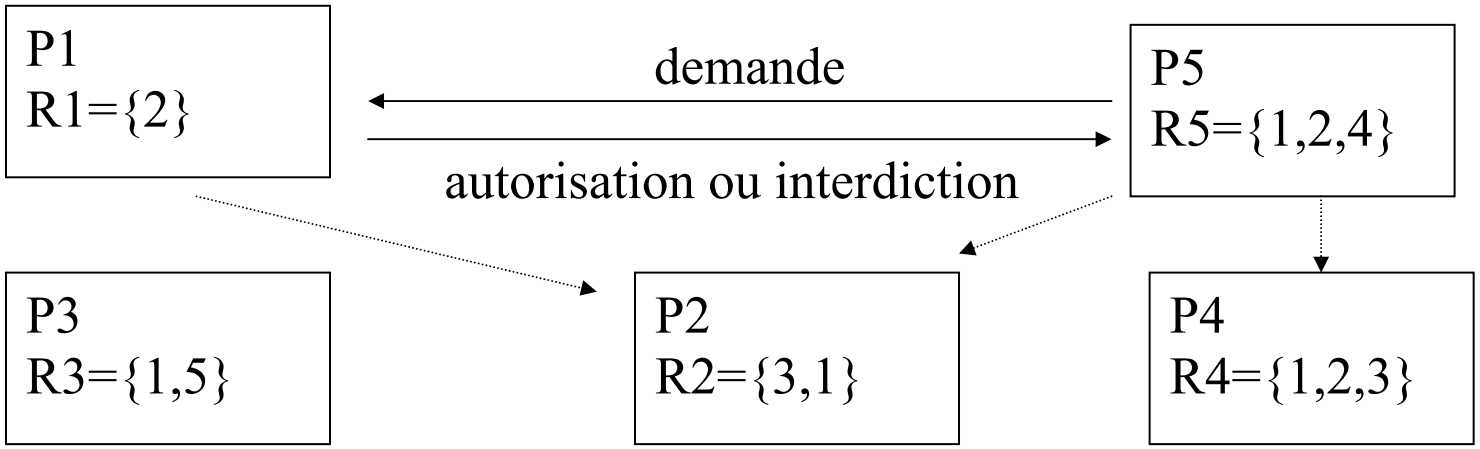
  - Algorithmes fondés sur un jeton
    Mouvement perpétuel :
    + : simplicité
    - : temps d'attente

    Compromis : diffusion du jeton.

♦ *Algorithmes fondés sur les permissions.*

Notation : i est un processus ayant besoin de la ressource
Ri est l'ensemble des processus à qui i doit demander la permission

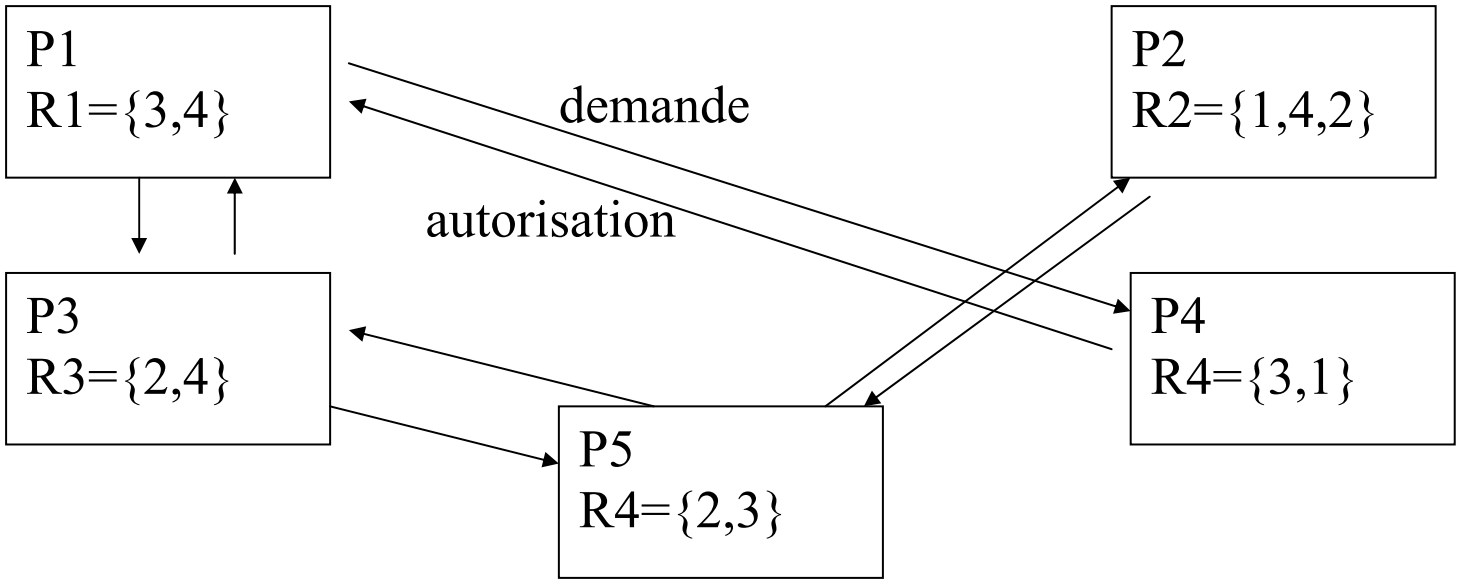
- *Permissions individuelles :*



P1 et P5 veulent la section critique → exclusion mutuelle car P5 demande à P1 donc P1 ne prendra pas la section critique s'il a autorisé P5.

Sûreté : $\forall$(i,j) : i$\in$Rj ou j$\in$Ri

- *Permissions d'arbitres*



P1 et P5 veulent la section critique → exclusion mutuelle par « l'arbitre » P3, qui ne l'accordera qu'à l'un des deux.

Sûreté : $\forall$(i,j) : Rj$\cap$Ri $\neq\varnothing$