

Ressources et Interblocage

Plan du chapitre

- Ressources
- Introduction à l'interblocage
- Définition de l'interblocage
- Approche informelle
- Modélisation de l'interblocage
- Détection-guérison de l'interblocage
- Prévention de l'interblocage
- Compléments : famine, verrouillage à 2 phases

Ressources

- Exemples
 - Imprimante, traceur, lecteur CD, table
- Allocation, libération
 - Une allocation entraîne un blocage du processus si la ressource demandée est allouée à un autre
- Réquisition (« preemption ») d'une ressource

Introduction à l'interblocage

- Illustrations
 - Carrefour
 - Chasse-neige
- En informatique
 - Exclusions mutuelles « croisées »

Définition

- Un ensemble de processus sont interbloqués si chacun d'eux est bloqué en attente d'un événement qui ne peut être déclenché que par un autre processus de l'ensemble

Étude informelle (1)

- Représentation graphique de l'exécution de deux processus
- Zones d'exclusion pour l'utilisation de ressources

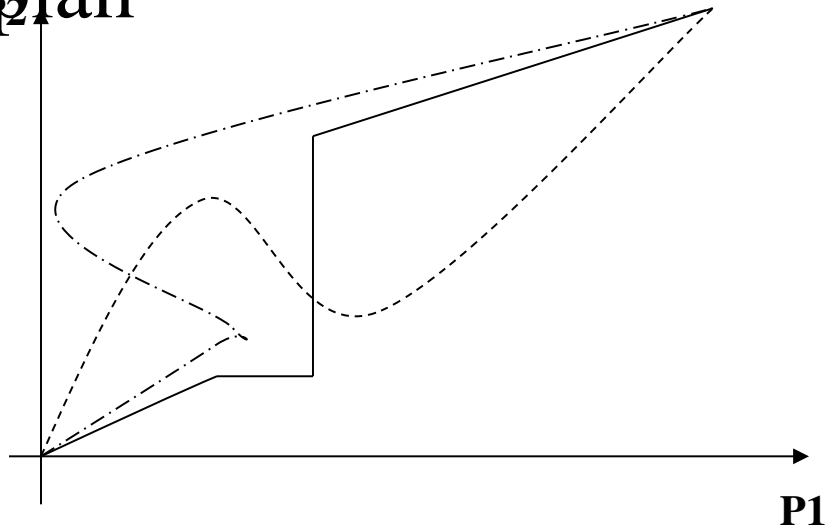
Etude informelle (2)

- Exécution d'un processus $P1(t)$: nombre d'instructions qui ont été exécutées à l'instant t
- $P1$ fonction non décroissante du temps

Etude informelle (3)

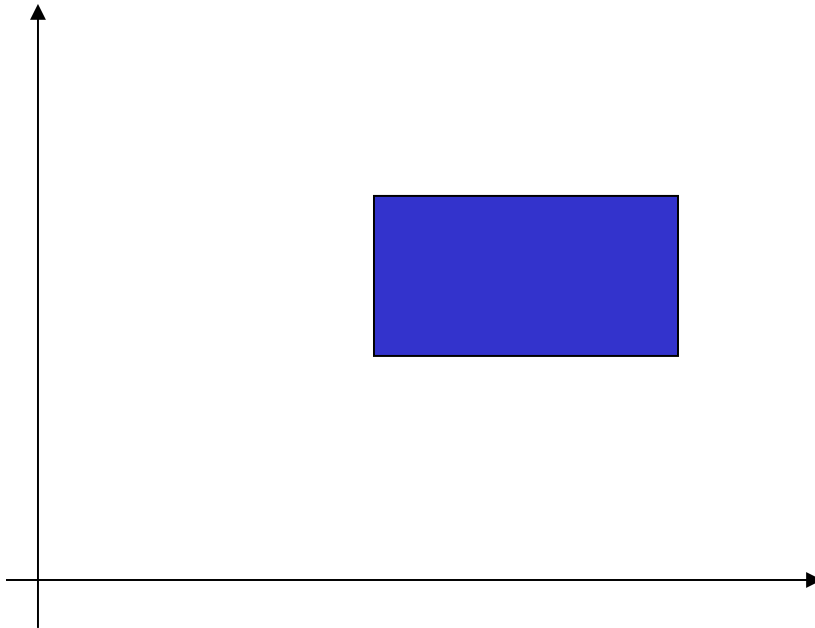
- L'exécution simultanée de 2 processus peut être représentée par une courbe dans un

plan



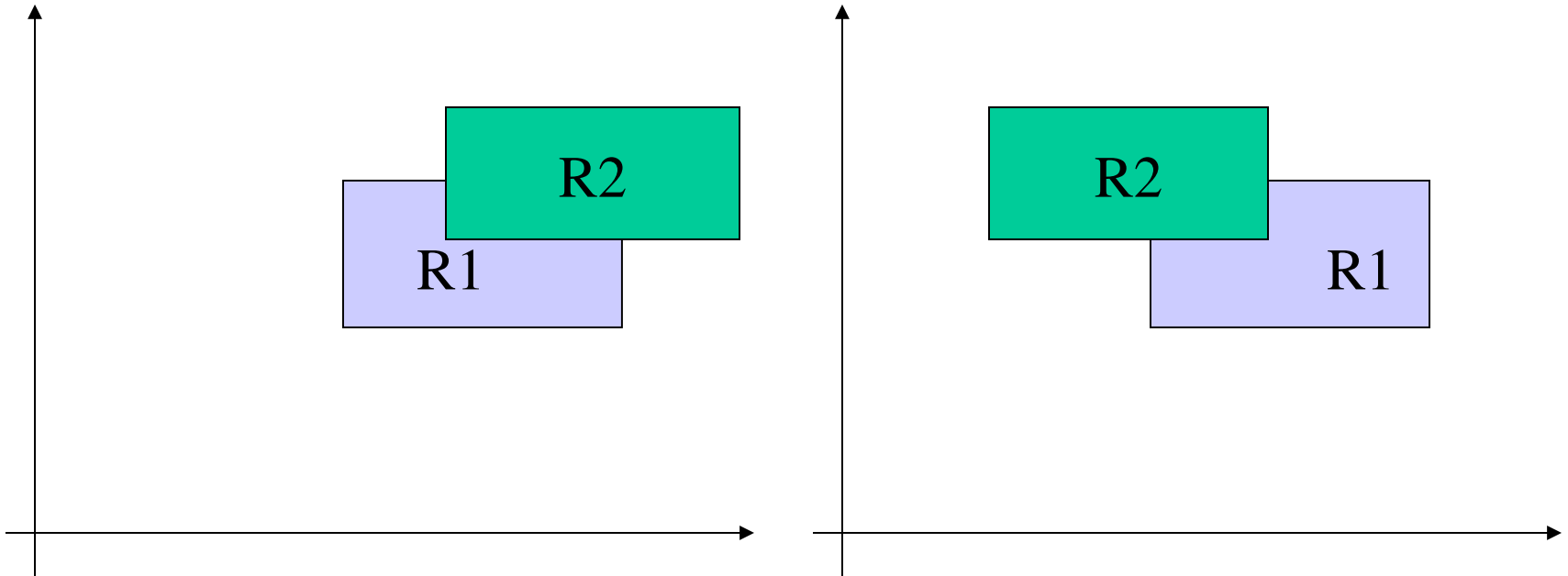
Etude informelle (4)

- Utilisation d'une ressource exclusive
- Une courbe d'exécution ne peut traverser le rectangle d'utilisation



Etude informelle (5)

- Utilisation de deux ressources

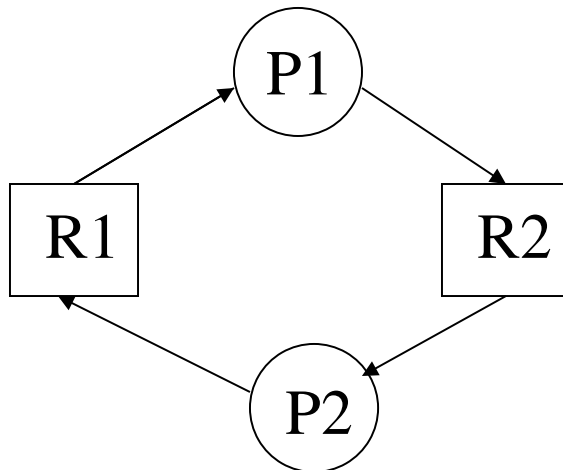


Conditions pour l'interblocage

- Exclusion mutuelle aux ressources
- Un processus qui possède des ressources peut faire de nouvelles demandes, et **se bloquer** en attente d'allocation
- Pas de réquisition
- Attente circulaire

Modélisation par graphe

- Nœuds de type ressource ou processus
- Arc ressource vers processus = allocation
- Arc processus vers ressource = demande
- Interblocage = cycle dans le graphe



Modélisation par matrices et vecteurs

- Ressources existantes, disponibles
- Matrice des allocations
- Matrice des demandes

4	2	3	1
---	---	---	---

E

2	1	0	0
---	---	---	---

D

allocations

0	0	1	0
2	0	0	1
0	1	2	0

demandes

2	0	0	1
1	0	1	0
2	1	0	0

Algorithme de l'autruche

- Ignorer purement et simplement les interblocages
- Réinitialiser en cas d'erreur

Détection de l'interblocage

- Recherche d'un cycle dans le graphe
- Représentation par matrice
 - Recherche d'un processus dont la demande peut être satisfaite
 - Terminer ce processus et libérer ses ressources
 - Recommencer jusqu'à ce que la liste des processus soit épuisée ou que il y ait impossibilité

Guérison de l'interblocage

- Suppression de processus
 - Lesquels ?
 - Est-il possible de détruire un processus pour le ré-exécuter à l'identique ?
 - Points de reprise
- Réquisition de ressources

Prévention de l'interblocage

- Nécessite des informations sur les demandes ultérieures des processus
- Algorithme classique : algorithme du banquier
 - Un processus doit annoncer au départ le nombre maximum de chaque ressource dont il aura besoin pour s'exécuter
 - On vérifie avant toute allocation qu'il y aura moyen de satisfaire les demandes max de tous les processus

Algorithme du banquier (ressource unique)

P1	0	6
P2	0	5
P3	0	4
P4	0	7

Libre = 10

P1	1	6
P2	1	5
P3	2	4
P4	4	7

Libre = 2

P1	1	6
P2	2	5
P3	2	4
P4	4	7

Libre = 1

Algorithme du banquier

Ressources multiples

- Ressources existantes, disponibles
- Matrice des allocations
- Matrice des annonces

4	2	3	1
---	---	---	---

E

2	1	0	0
---	---	---	---

D

Allocations

0	0	1	0
2	0	0	1
0	1	2	0

Annonce

3	2	1	1
2	1	0	1
3	1	2	1

Conditions suffisantes pour éviter l'interblocage

- Allouer toutes les ressources en une fois
- Libérer toutes les ressources avant une nouvelle demande
- Méthode des classes ordonnées
 - Ressources organisées en classes C_1, C_2, \dots, C_n
 - Dans une classe, allocation en bloc
 - Allocation dans l'ordre des classes

Verrouillage à deux phases

- Très utilisé en bases de données
- Ressources inconnues au départ
- Première phase : acquisition des ressources
- Deuxième phase : utilisation, puis libération
- En cas d'échec dans la première phase (demande d'une ressource déjà occupée), les ressources allouées au processus sont libérées et on recommence.

Famine

- Exemple : les philosophes aux spaghetti
 - 2 philosophes qui mangent en alternance peuvent empêcher un autre de manger
- Prévenir la famine revient à allouer les ressources de façon telle qu'une demande soit satisfaite au bout d'un temps fini

Équité

- Garantir que tous les processus de même priorité sont servis de façon comparable
- Différence avec la famine