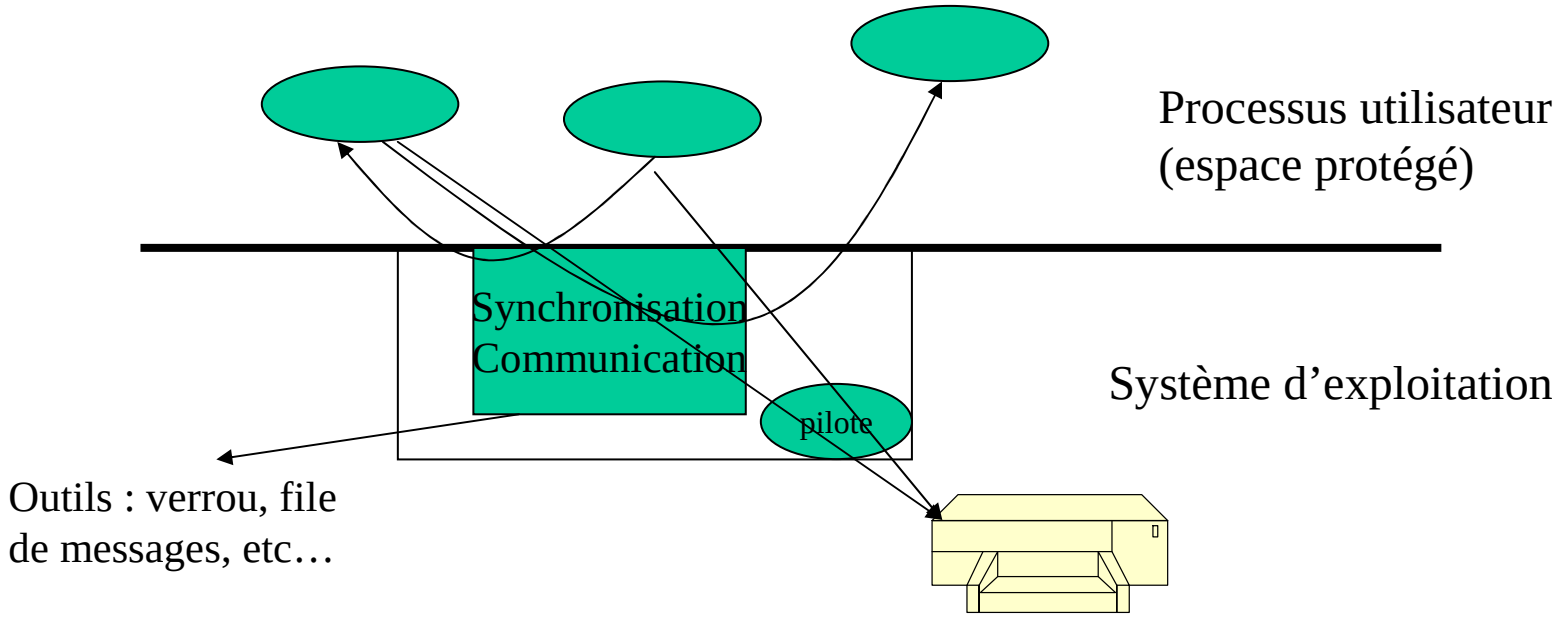
Synchronisation et communication entre processus

Principes :Exclusion mutuelle et interblocage
Communication entre processus



INTRODUCTION

- **Nous nous intéressons au développement d'applications multiprocessus concurrents c'est à dire d'applications composées de plusieurs processus indépendants et en concurrence pour l'accès aux ressources du système.**

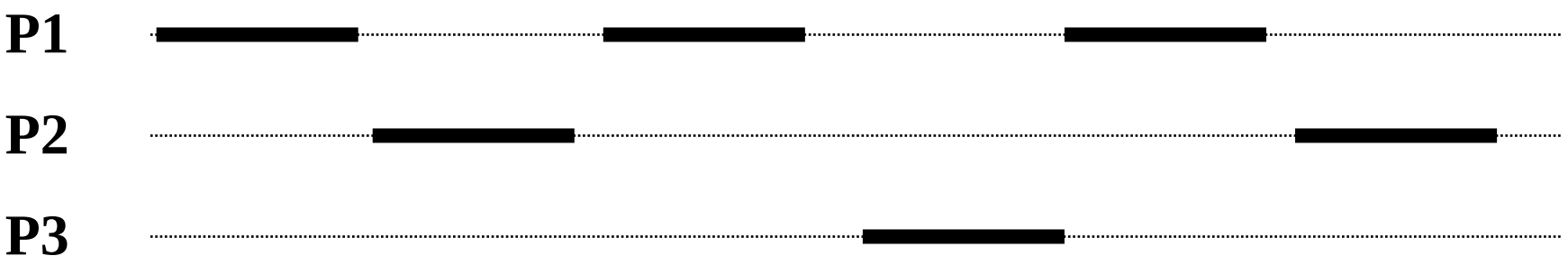
Les processus sont ordonnancés indépendamment les uns des autres.

Une ressource désigne toute entité dont a besoin un processus pour s'exécuter.

- Ressource matérielle (processeur, périphérique)
- Ressource logicielle (fichier)

INTRODUCTION

- **Système multiprocessus**
- **L'ordonnancement "entrelace" les exécutions**



☞ **Processus non indépendants :**
accès concurrents aux ressources

Notion de ressources

- Définitions

  – Une ressource désigne toute entité dont a besoin un processus pour s'exécuter.
    - Ressource matérielle (processeur, périphérique)
    - Ressource logicielle (fichier, variable).

  – Une ressource est caractérisée
    - par un état : libre / occupée
    - par son nombre de points d'accès (nombre de processus pouvant l'utiliser en même temps)

Notion de ressources

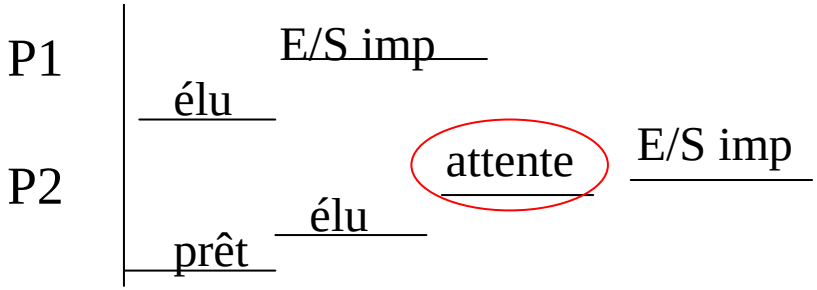
- Utilisation d'une ressource par un processus

  – Trois étapes : Allocation
  Utilisation
  Restitution

  – Les phases d'allocation et de restitution doivent assurer que le ressource est utilisée conformément à son nombre de points d'accès

    - ressource critique à un seul point d'accès

Notion de ressources
Exemple

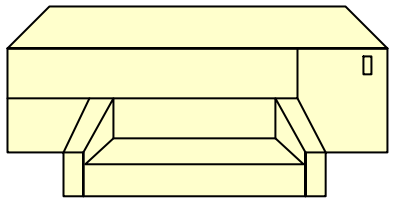


- Ressource matérielle : imprimante

Processus P1

Print (texte)

SE
Gestion des entrées/sorties
PRINT



LIBRE
1 point d'accès

Print (texte)

Allouée P1

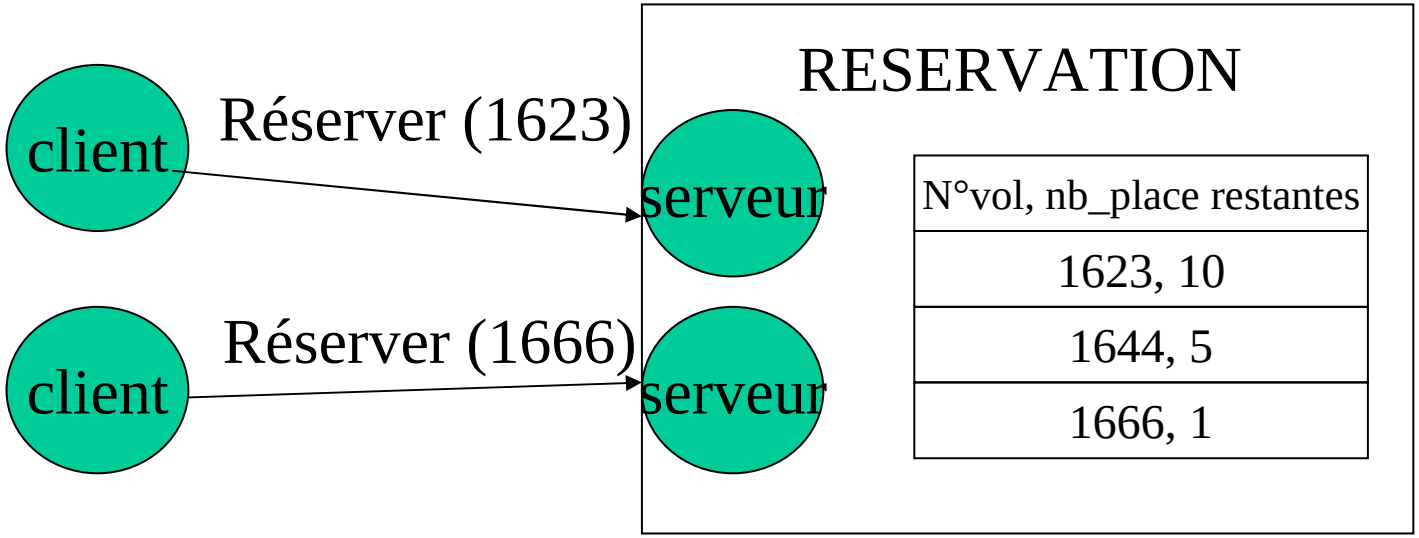
OCCUPEE
1 point d'accès

Processus P2

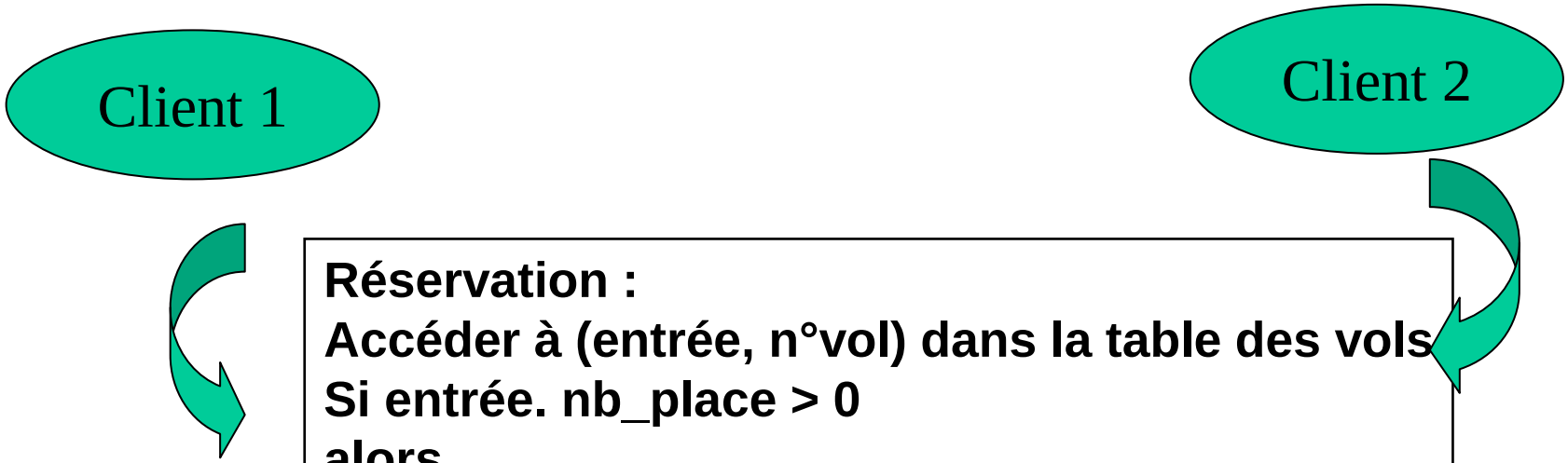
P2 en attente jusqu'à
libération par P1

I.
Un système de réservation : Notion d'exclusion mutuelle entre processus

- On considère un système permettant à des clients de réserver une place dans un avion donné numéro_vol.

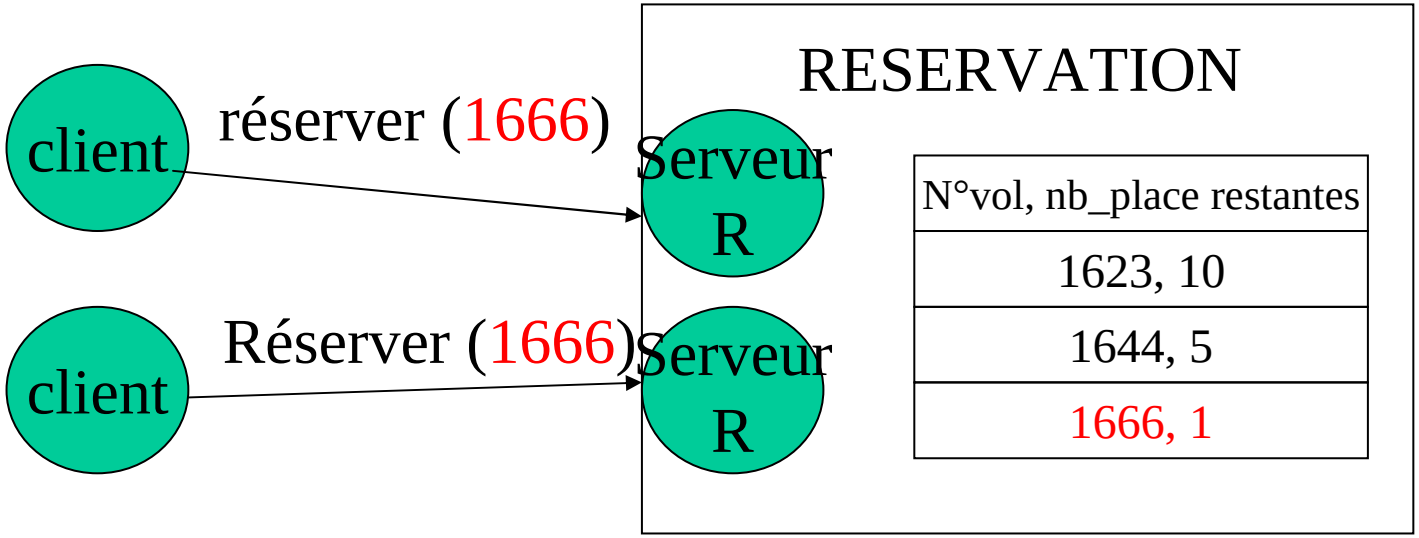


Exclusion mutuelle entre processus



Un système de réservation : Notion d'exclusion mutuelle entre processus

- On considère la situation où deux clients demandent simultanément la réservation d'une place sur le vol 1666 pour lequel reste une seule place disponible

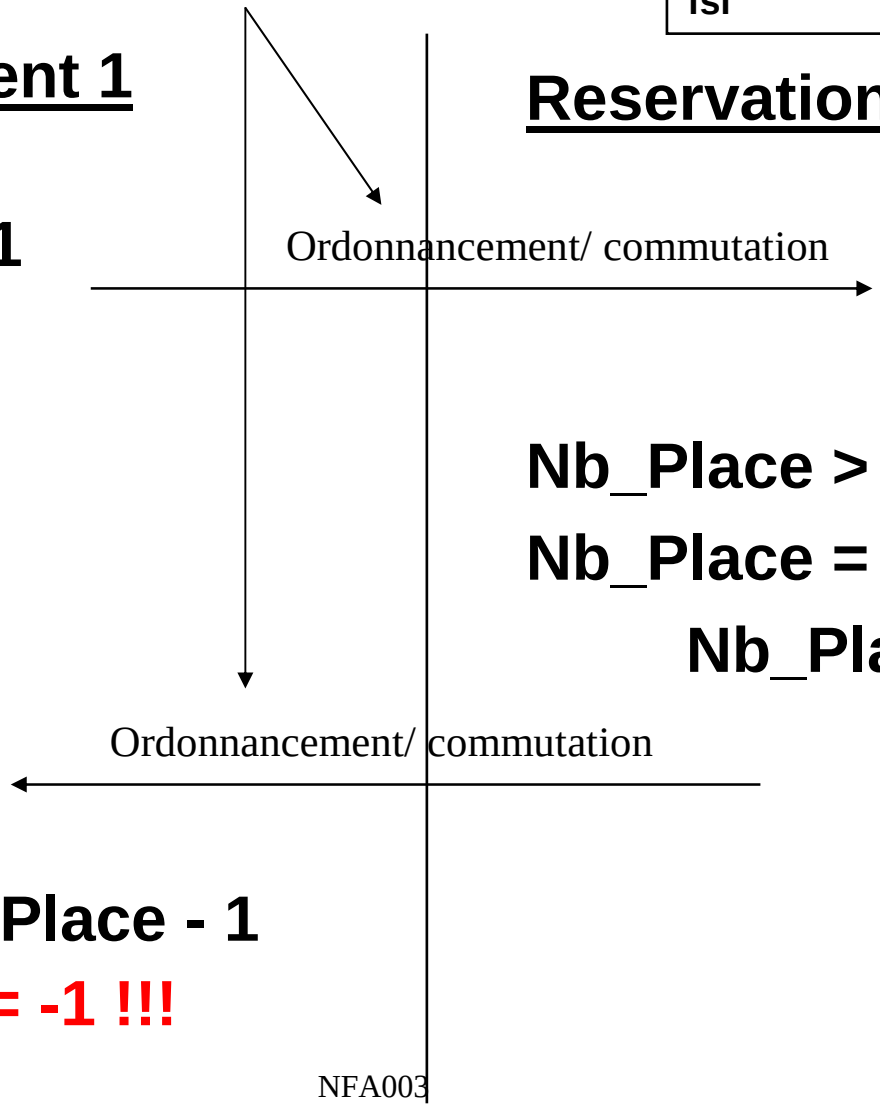


➢ Les deux processus Réservation s'exécutent en concurrence ; le SE ordonnance les deux processus via un algorithme en temps partagé

Réservation :
Si nb_place > 0
alors
Réserver une place
nb_place = nb_place - 1
fsi

Réservation Client 1

Nb_Place > 0 = 1



Reservation Client 2

Nb_Place > 0 = 1

Nb_Place = Nb_Place - 1

Nb_Place = 0

Nb_Place = Nb_Place - 1

Nb_Place = -1 !!!

➢ Que s'est-il passé ? L'entrelacement des exécutions a permis à Réservation_Client_2 de modifier nb_place alors que Réservation_Client_1 accédait déjà à cette variable

Réservation :
Si nb_place > 0
alors
Réserver une place
nb_place = nb_place - 1
fsi

Réservation Client 1

Reservation Client 2

Nb_Place > 0 = 1

Ordonnancement/ commutation

Réservation client 1 a mémorisé la valeur de nb_place

Nb_Place > 0 = 1

Réservation client 2 modifie nb_place

Nb_Place = Nb_Place - 1

Nb_Place = 0

Réservation client 1 modifie à son tour nb_place sans retester sa valeur

Ordonnancement/ commutation

Nb_Place = Nb_Place - 1

Nb_Place = -1 !!!

➢ Que faut-il faire ? Interdire au processus Reservation_Client_2 de modifier la variable nb_place pendant que Reservation_Client 1 manipule cette variable
→ un seul processus à la fois accède à nb_place.
➢Nb_Place est une ressource critique à un seul point d'accès.

Reservation Client 1

```
Si nb_place > 0
alors
    Réserver une place
    nb_place = nb_place - 1
fsi
```

Reservation Client 2

Nb_Place non accessible
Pour le client 2 tant que
client 1 manipule cette variable

Section critique et exclusion mutuelle

Processus
Début

Entrée Section Critique

Ressource Critique
Nb_Place

Sortie Section Critique

Fin

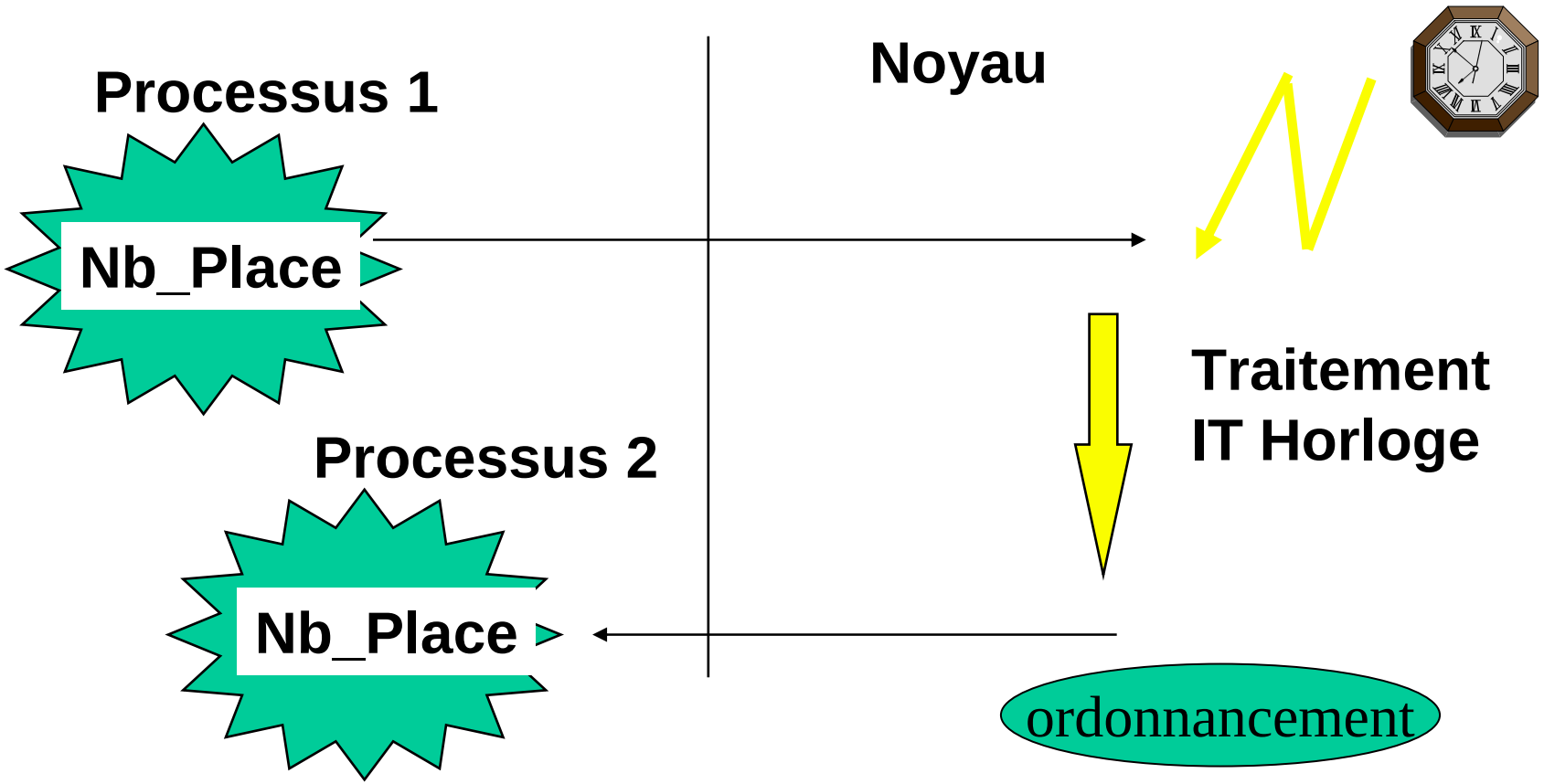
- **Ressource utilisable par un seul processus à la fois**

SECTION CRITIQUE (code d'utilisation de la ressource critique)

☞ L'entrée et la sortie de SC doivent assurer qu'à tout moment, un seul processus s'exécute en SC (exclusion mutuelle)

Notion d'exclusion mutuelle entre processus

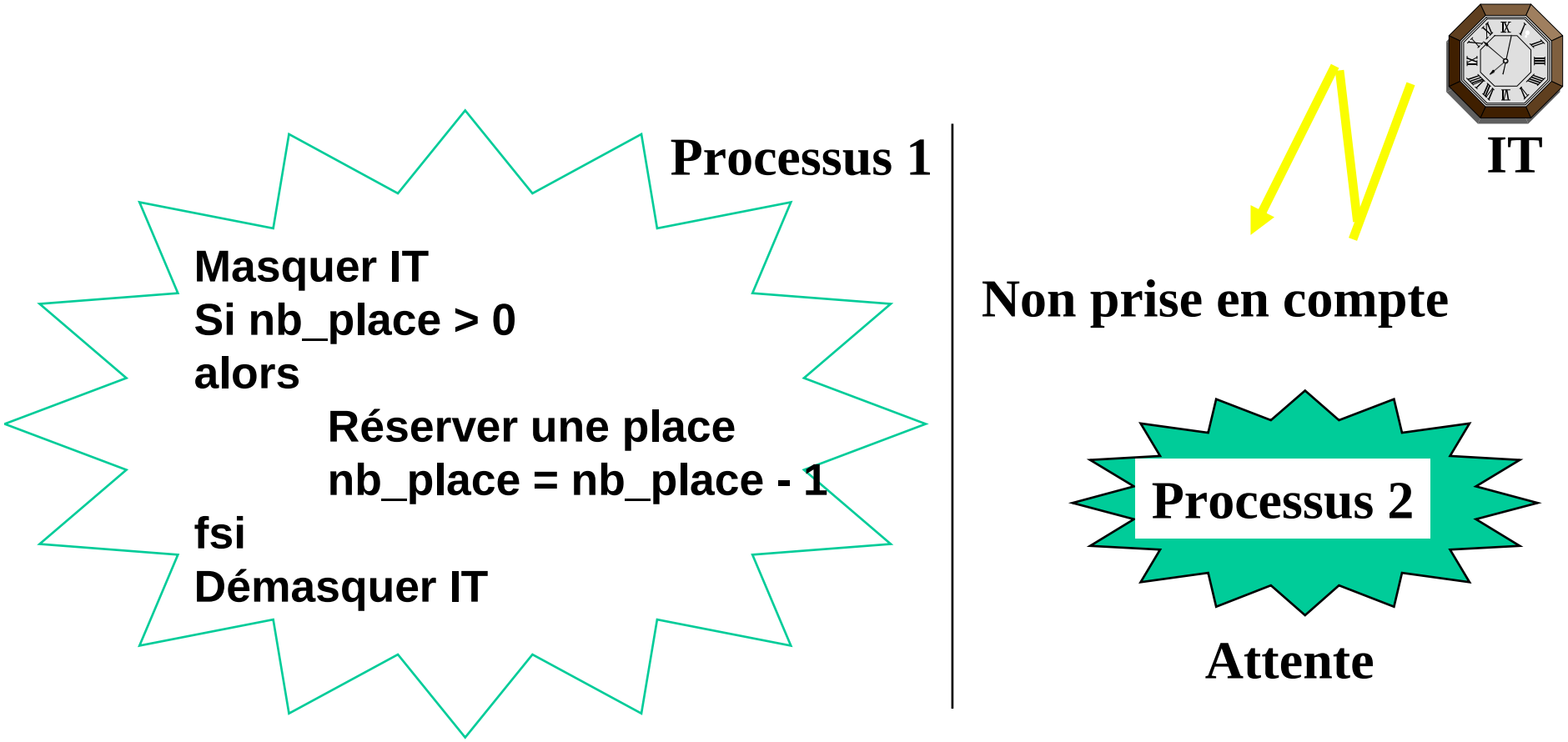
• Le processus Réservation_Client_2 a pu s'exécuter car l'ordonnanceur a préempté Réservation_Client_1 et élu Réservation_Client_2.



Notion d'exclusion mutuelle entre processus
Solution matérielle

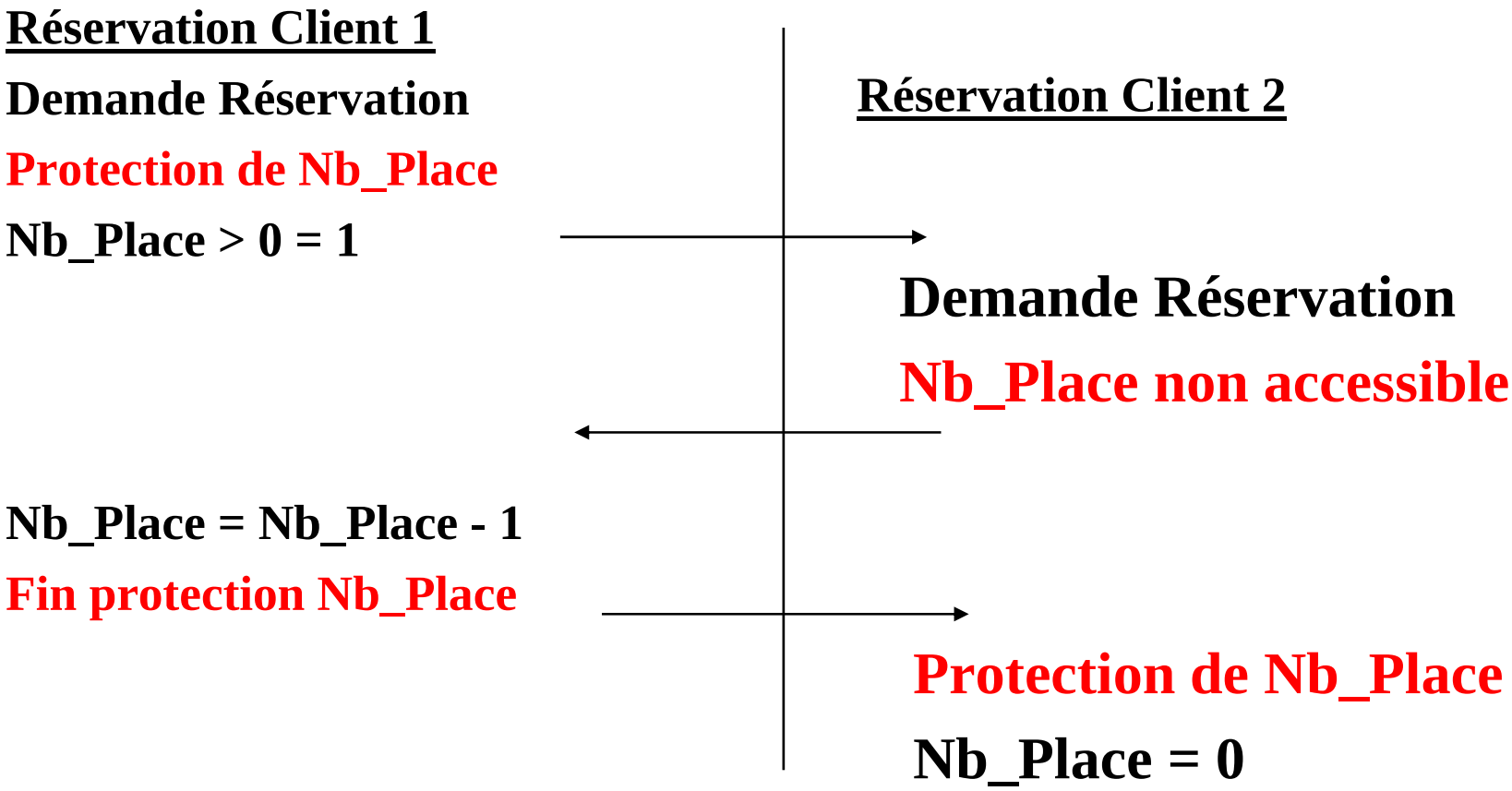
Une première solution pour empêcher Réservation_Client_2 de s'exécuter est de masquer les interruptions

Mais mode superviseur



Notion d'exclusion mutuelle entre processus
Solution logicielle

On ne souhaite plus interdire l'entrelacement des processus.
On protège l'accès à Nb_Place : le processus Réservation_Client_1
« verouille » l'accès à Nb_Place tant qu'il utilise la variable



Notion d'exclusion mutuelle entre processus
Solution logicielle : le verrou

- Un mécanisme proposé pour permettre de résoudre l'exclusion mutuelle d'accès à une ressource est le mécanisme de *verrou.* Un verrou est un objet système à deux états (libre/occupé) sur lequel deux opérations sont définies..
  - *verrouiller (v)* permet au processus d'acquérir le verrou *v* s'il est libre. S'il n'est pas disponible, le processus est bloqué en attente de la ressource.
  - *déverrouiller (v)* permet au processus de libérer le verrou *v* qu'il possédait. Si un ou plusieurs processus étaient en attente de ce verrou, un seul de ces processus est réactivé et reçoit le verrou.
- En tant qu'opérations systèmes, ces opérations sont indivisibles, c'est-à-dire que le système qu'elles s'exécutent interruptions maquées.

Notion d'exclusion mutuelle entre processus
Solution logicielle : le verrou

Réservation Client 1

V_Nb_Place : verrou;

Demande Réservation

Protection de Nb_Place →

Nb_Place > 0 = 1

Verrouiller (V_Nb_Place)
Si V_Nb_Place libre alors autoriser l'accès et mettre V_Nb_Place à l'état occupé sinon bloquer le processus

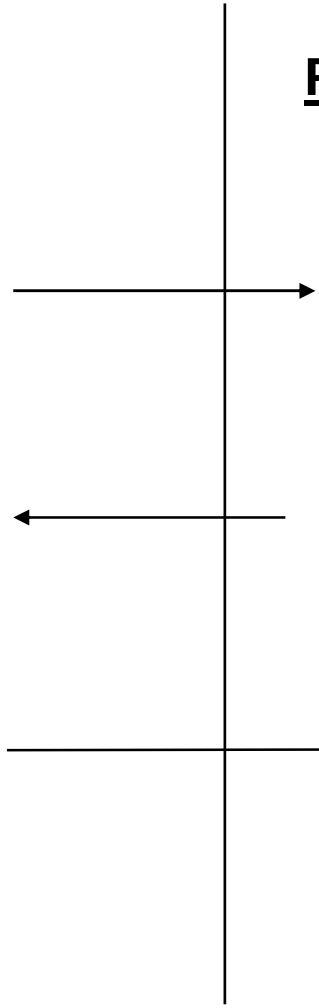
Nb_Place = Nb_Place - 1

Fin protection Nb_Place →

Déverrouiller (V_Nb_Place)
Si un processus bloqué en attente pour accéder à Nb_Place, le débloquer, Sinon V_Nb_Place libre

Notion d'exclusion mutuelle entre processus
Solution logicielle

V_Nb_Place : verrou; -- verrou libre



Réservation Client 1

Demande Réservation

Verrouiller (V_Nb_Place)

V_Nb_Place libre

Nb_Place > 0 = 1

Nb_Place = Nb_Place - 1

Déverrouiller (V_Nb_Place)

Réveil de réservation_client_2

Réservation Client 2

Demande Réservation

Verrouiller (V_Nb_Place)

V_Nb_Place occupé par client_1

Réservation_Client _2 bloqué

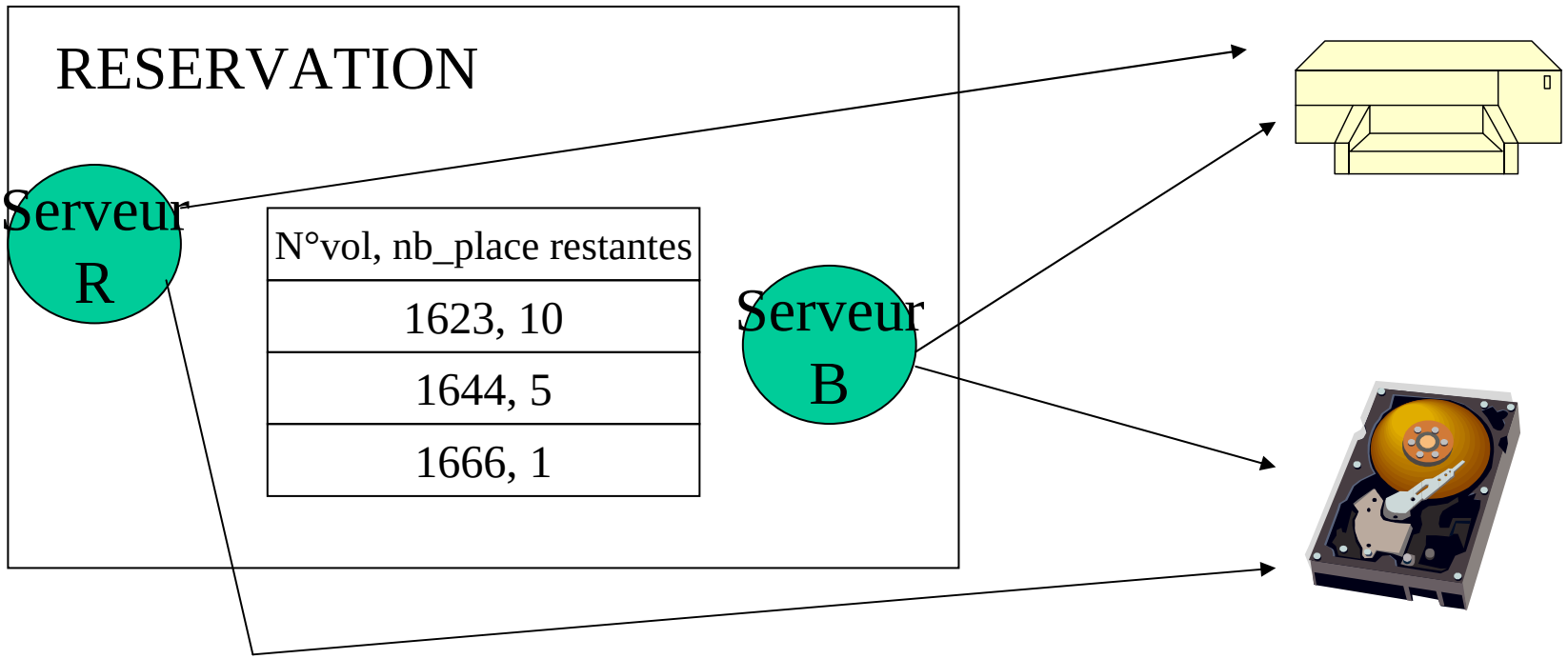
Nb_Place = 0

Déverrouiller (V_Nb_Place)

V_Nb_Place : état libre

II.
Un système de réservation : Notion d'interblocage

- On considère un système permettant à des clients de réserver une place dans un avion donné numéro_vol.



Notion d’interblocage

- Le système de réservation dispose de deux ressources : une imprimante, un disque

- Le processus Reservation_client, une fois la réservation effectuée, imprime une facture pour le client et écrit sur le disque dans un fichier commande, un enregistrement correspondant (commande en cours, paiement en attente)

- Un processus Billet, régulièrement, parcourt dans le fichier commande, les enregistrements commande et pour chacun d’eux pour lesquels un paiement a été reçu, passe la commande à état payé édite sur l’imprimante un billet.

Notion d'interblocage

- On programme comme suit les deux processus

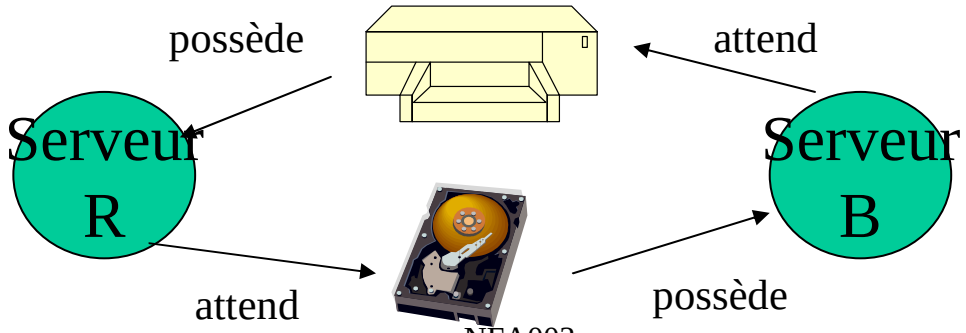
Processus Réservation client	Processus Billet
Effectuer réservation client Verrouiller (Imprimante) Verrouiller (Fichier Commande) Imprimer (facture) Ecrire (commande, Fichier Commande) Deverrouiller (Imprimante) Deverrouiller (Fichier Commande)	Toutes les P unités de temps Faire Verrouiller (Fichier Commande) Verrouiller (Imprimante) Tant que (commandes) Si paiement mettre état payé Imprimer (billet) Fsi Fin tant que Deverrouiller (Fichier Commande) Deverrouiller (Imprimante)

Notion d'interblocage

Processus Réservation client (RC)	Processus Billet (B)
Effectuer réservation client Verrouiller (Imprimante) (imprimante libre allouée à RC) Verrouiller (Fichier Commande) (RC Bloqué)	 REVEIL BILLET Verrouiller (Fichier Commande) (fichier commande libre, alloué à B) Verrouiller (Imprimante) (B bloqué)
Imprimer (facture) Ecrire (commande, Fichier Commande) Deverrouiller (Imprimante) Deverrouiller (Fichier Commande)	Tant que (commandes) Si paiement mettre état payé Imprimer (billet) Fsi Fin tant que Deverrouiller (Fichier Commande) Deverrouiller (Imprimante)

Notion d'interblocage

Processus Réservation client (RC)	Processus Billet (B)
Effectuer réservation client Verrouiller (Imprimante) (imprimante libre allouée à RC)	REVEIL BILLET Verrouiller (Fichier Commande) (fichier commande libre, alloué à B Verrouiller (Imprimante) (B bloqué)
Verrouiller (Fichier Commande) (RC Bloqué)	INTERBLOCAGE



Notion d'interblocage

- **Les deux processus demandent l'accès aux ressources dans un ordre différent.**
- **→ imposer un ordre de demande des ressources**

PR
Verrouiller (Imprimante)
Verrouiller (Fichier Commande)
…
Deverrouiller (Imprimante)
Deverrouiller (Fichier Commande)

B
Verrouiller (Fichier Commande)
Verrouiller (Imprimante)
…
Deverrouiller (Imprimante)
Deverrouiller (Fichier Commande)

B
Verrouiller (Imprimante)
Verrouiller (Fichier Commande)
…
Deverrouiller (Imprimante)
Deverrouiller (Fichier Commande)

Notion d'interblocage

Processus Réservation client (RC)	Processus Billet (B)
Effectuer réservation client Verrouiller (Imprimante) (imprimante libre allouée à RC)	REVEIL BILLET Verrouiller (Imprimante) (B bloqué)
Verrouiller (Fichier Commande) (Fichier libre alloué à RC)	
Imprimer (facture) Ecrire (commande, Fichier Commande)	BILLET débloqué, acquiert Fichier Commande et s'exécute
Deverrouiller (Imprimante) Deverrouiller (Fichier Commande)	