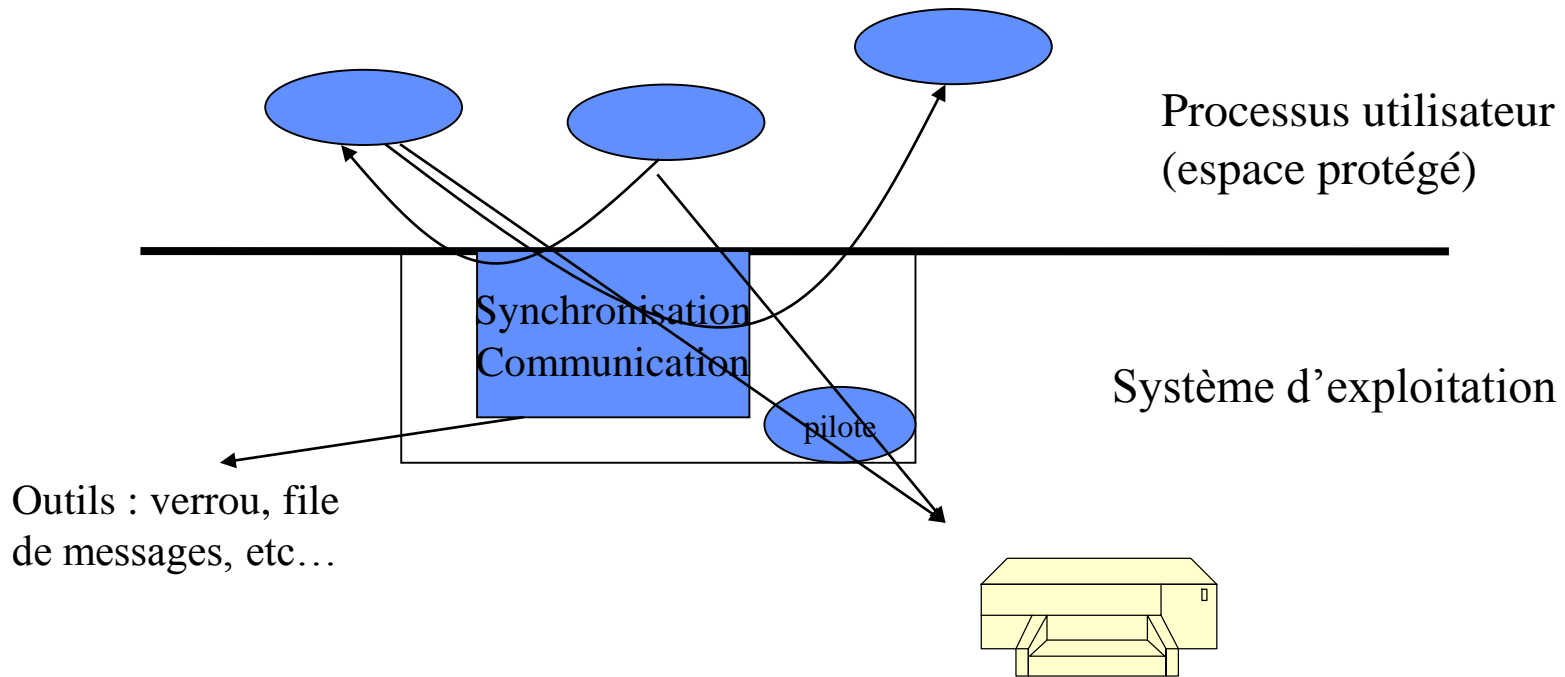


# Synchronisation entre processus



# Synchronisation et communication entre processus

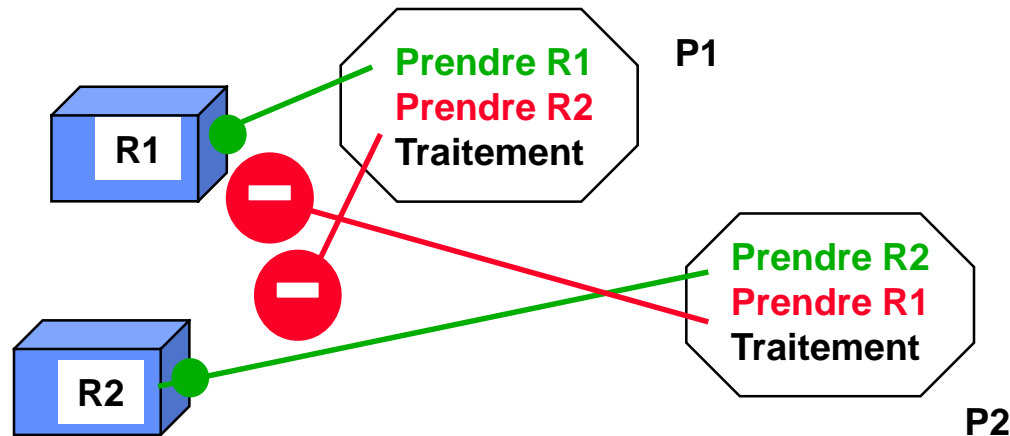
Interblocage et coalition

# Interblocage, Famine et Coalition

- Interblocage

Ensemble de n processus attendant chacun une ressource déjà possédée que par un autre processus de l'ensemble

R1 et R2 à un seul point d'accès



☞ **Aucun processus ne peut poursuivre son exécution**  
**Attente Infinie**

# Un exemple d'interblocage

- CLIENT

```
/* ouverture du tube tube1 en écriture */
```

```
tub1 = open ("tube1", O_WRONLY); -- en attente de l'ouverture en  
lecture de tube1
```

```
/* ouverture du tube tube2 en lecture */
```

```
tub2 = open ("tube2", O_RDONLY);
```

- 

- SERVEUR

```
/*ouverture du tube 2 en écriture */
```

```
tub2 = open ("tube2", O_WRONLY); -- en attente de l'ouverture en  
lecture de tube2
```

```
/* ouverture du tube 1 en lecture */
```

```
tub1 = open ("tube1", O_RDONLY);
```

# Conditions nécessaires à l'obtention d'un interblocage

- **Exclusion mutuelle**
  - Une ressource au moins doit se trouver dans un mode non partageable
- **Occupation et attente**
  - Un processus au moins occupant une ressource attend d'acquérir des ressources supplémentaires détenues par d'autres processus
- **Pas de réquisition**
  - Les ressources sont libérées sur seule volonté des processus les détenant
- **Attente circulaire**

# Attente circulaire

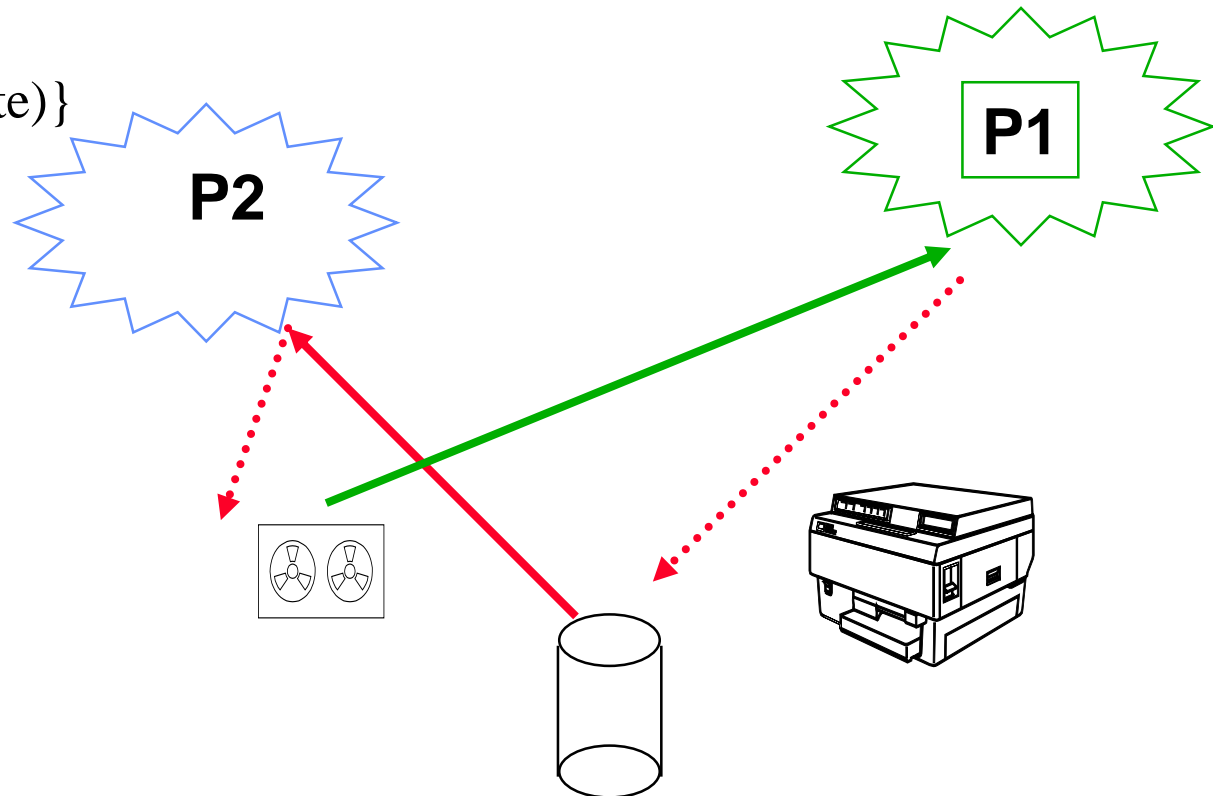
P1 {	P2 {
P(bande)	P(disque)
P(disque)	P(bande)
.....	.....
V(bande)	V(bande)
V(disque)	V(disque)
P(imprimante)	P(imprimante)
....	
V(imprimante)}	V(imprimante)}

**P1 : bande**

**P2 : disque**

**P2 : attente bande**

**P1 : attente disque**



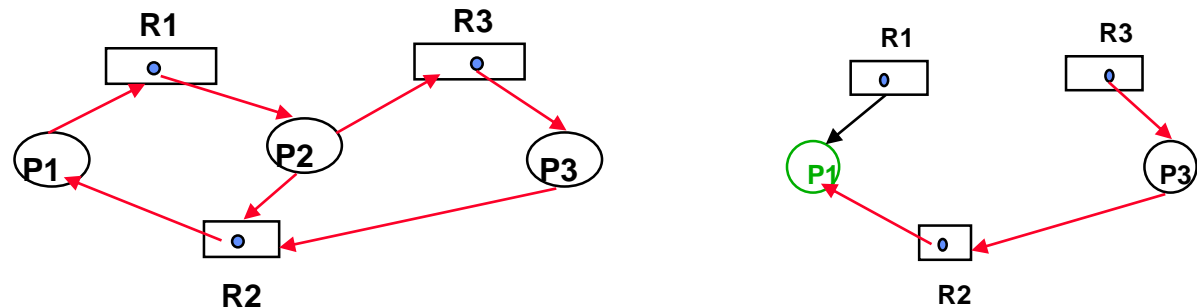
# Méthodes de traitement des interblocages

- **Permettre l'interblocage et le corriger**
  - ☞ **Politique de guérison**
- **Ne pas permettre l'interblocage**
  - ☞ **Politique de prévention ou d'évitement**
- **Ignorer le problème**
  - ☞ **Politique de l'Austruche**

# Politiques de guérison

- Le système maintient un graphe représentant l'allocation des ressources et les attentes des processus
- Régulièrement, le système parcourt le graphe à la recherche de cycles
- Si un cycle est découvert, celui-ci est cassé en avortant les processus en interblocage appartenant au cycle

👉 coûteux





# Politiques de prévention

- **Assurer qu'au moins une des conditions nécessaires ne peut avoir lieu**
  - ☞ **Exclusion mutuelle : difficile**
  - ☞ **Occupation et attente : demander les ressources en une seule fois**
  - ☞ **Pas de réquisition : difficile**
  - ☞ **Attente circulaire : ordre total sur l'ordre de demandes de ressources**

# Politiques de prévention

- Occupation et attente : demander les ressources en une seule fois

```
P1 {  
P(bande)  
P(disque)  
.....  
V(bande)  
V(disque)  
P(imprimante)  
....  
V(imprimante)}
```

```
P2 {  
P(disque)  
P(bande)  
.....  
V(bande)  
V(disque)  
P(imprimante)  
....  
V(imprimante)}
```



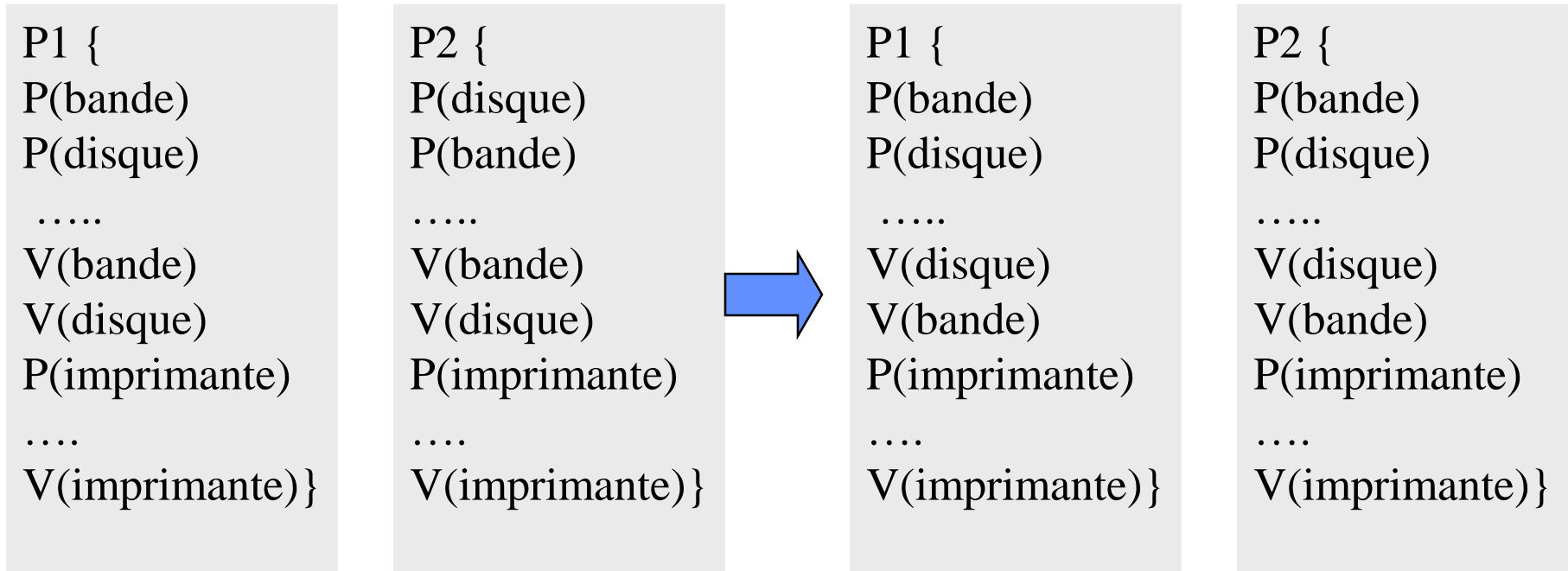
```
P2 {  
P(disque, bande, imprimante)  
.....  
V(bande)  
V(disque)  
V(imprimante)}
```

```
P1 {  
P(bande, imprimante, disque)  
....  
V(bande)  
V(disque)  
V(imprimante)}
```

↪ Mauvaise utilisation des ressources

# Politiques de prévention

- **Attente circulaire** : ordre total sur l'ordre de demandes de ressources
- **Unité de bandes avant le disque et avant l'imprimante**



# Politiques d'évitement

- Examen dynamique de l'état d'allocation des ressources afin d'éviter l'attente circulaire

- 👉 A chaque demande d'allocation de ressource, le système détermine si accepter cette allocation peut ou non mener le système à l'interblocage, ie l'état du système reste-t-il *sain*.

- 👉 si oui, l'allocation est refusée.

- 👉 Vision pessimiste

# Politiques d'évitement

	Besoins maximaux	Ressources allouées	Demandes restantes
P1	10	5	5
P2	4	2	2
P3	9	2	7

12 exemplaires de ressources au total

Le nombre de ressources disponibles est égal à 3.

La séquence d'exécution  $\langle P2, P1, P3 \rangle$  est saine:

- satisfaction de P2, ressources disponibles = 1;
- restitution des ressources par P2, ressources disponibles = 5;
- satisfaction de P1, ressources disponibles = 0;
- restitution des ressources par P1, ressources disponibles = 10;
- satisfaction de P3, ressources disponibles = 3;
- restitution des ressources par P3, ressources disponibles = 12.

# Politiques d'évitement

	Besoins maximaux	Ressources allouées	Demandes restantes
P1	10	5	5
P2	4	2	2
P3	9	<b>3</b>	6

12 exemplaires de ressources au total

Le nombre de ressources disponibles est égal à 2

L'état devient malsain et aucune séquence d'exécution incluant les trois processus ne peut être construite. Ici, seul P2 peut être satisfait:

- satisfaction de P2, ressources disponibles = 0;
- restitution des ressources par P2, ressources disponibles = 4.

Maintenant, ni P3, ni P2 ne peuvent être satisfaits.

# Politique de l'Autruche

- **Prétendre que les interblocages ne se produisent jamais et **ne rien prévoir****
  - ☞ **Un interblocage peut se produire et n'est pas détecté**
  - ☞ **Détérioration des performances jusqu'à arrêt complet de l'application**
  - ☞ **Rédémarrage manuel de l'application**

# Politique de l'Autruche

- **Justification de ce choix**

**Fréquence de l'interblocage**

