

# 9.4

## Les interblocages

NSI TERMINALE - JB DUTHOIT

### Exercice 9.11

Un système utilise un ordonnancement Tourniquet (Round-Robin) avec un quantum  $q = 2$ . Le système dispose de deux ressources distinctes :  $R_1$  et  $R_2$ . L'accès aux ressources se fait en exclusion mutuelle exclusive (une ressource ne peut être détenue que par un seul processus à la fois).

Deux processus A et B arrivent au temps  $t = 0$  (dans l'ordre A puis B dans la file).

- **Processus A (Durée totale 6 u.t.) :**

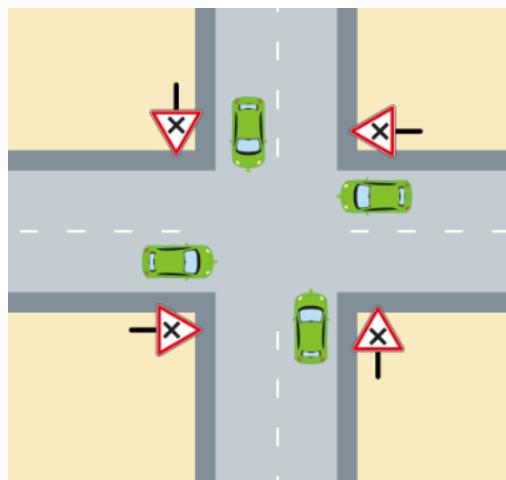
- À  $t_{exec} = 1$  : Demande  $R_1$ .
- À  $t_{exec} = 5$  : Demande  $R_2$ .
- Libère toutes ses ressources à la fin de son exécution.

- **Processus B (Durée totale 6 u.t.) :**

- À  $t_{exec} = 1$  : Demande  $R_2$ .
- À  $t_{exec} = 3$  : Demande  $R_1$ .
- Libère toutes ses ressources à la fin de son exécution.

1. Si un processus demande une ressource occupée, il se bloque immédiatement.
2. Il n'y a pas de mécanisme de détection ou de prévention des interblocages.
1. Tracez le diagramme de Gantt de l'exécution.
2. Que se passe-t-il précisément à l'instant  $t = 6$  ?
3. Comment appelle-t-on cette situation en informatique ?
4. Citez les quatre conditions nécessaires pour que cette situation arrive (Conditions de Coffman).

Les interblocages sont des situations de la vie quotidienne !



Qui doit passer ?

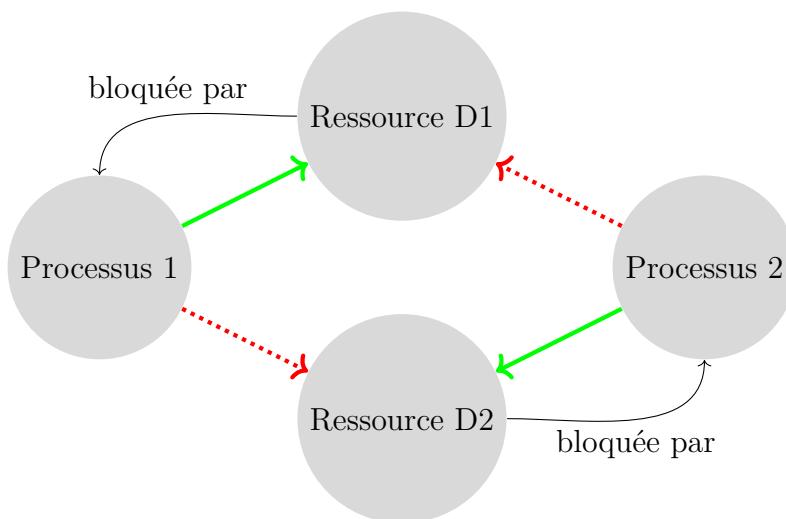
### 9.4.1 Quels interblocages en informatique ? Exemple de deux processus qui ont besoin de la même donnée de manière exclusive (pour la modifier par exemple)

Su le schéma ci-dessous, P1 et P2 ont tous les deux besoin des ressource D1 et D2  
Voici un scénario possible :

- Le processus P1 commence son exécution (état élu), il demande la ressource D1. Il obtient satisfaction puisque D1 est libre
- P1 passe ensuite l'état "prêt".
- Pendant ce temps, le système a passé P2 en exécution : P2 commence son exécution et demande la ressource D2. Il obtient immédiatement D2 puisque cette ressource était libre.
- P2 poursuit son exécution
- P2 demande la ressource D1, il se retrouve dans un état bloqué puisque la ressource D1 a été attribuée à P1
- P1 est dans l'état prêt, il n'a pas eu l'occasion de libérer la ressource R1 puisqu'il n'a pas eu l'occasion d'utiliser R1
- P1 passe donc à l'état d'exécution . Afin de libérer D1, P1 a besoin de D2. Problème, car D2 n'est pas disponible ! Donc P1 ne peut pas libérer D1 et passe à bloqué.

Résumons la situation à cet instant : P1 possède la ressource D1 et se trouve dans l'état bloqué, P2 possède la ressource D2 et se trouve dans l'état bloqué (attente de D1)

Cette situation est qualifiée d'**interblocage** (deadlock en anglais).



Face à cette problématique, deux situations sont envisageables :

- Essayer d'éviter les interblocages.
- Déetecter quand un interblocage est apparu, et le solutionner

☞ La plupart des systèmes d'exploitation ont choisi de ne pas essayer de les éviter, mais plutôt de détecter et de les solutionner.