

SYS2

Système d'exploitation

M.Bastreggi (mba)

Haute École Bruxelles Brabant — École Supérieure d'Informatique

Année académique 2020 / 2021

interblocages

interblocages

plusieurs processus → partagent la Mémoire, le CPU et aussi des **ressources**

exemples de **ressources** :

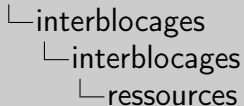
mémoires de masse, imprimante, graveur,
informations (enregistrement d'une BD), entrées
de la table des processus...

ressources

classification

Nous classons les ressources en

- ▶ **partageables** / non partageables
- ▶ et/ou
- ▶ **préemptibles** / non préemptibles

**classification**

Nous classons les ressources en

- **partageables** / non partageables
- et/ou
- **préemptibles** / non préemptibles

Ces propriétés sont liées à la nature des ressources, mais aussi à la manière dont ces ressources sont gérées.

Exemple : une imprimante est par sa nature **non partageable**, mais elle devient **partageable** par l'utilisation d'un spooler d'impression. (voir techniques de prévention d'interblocages)

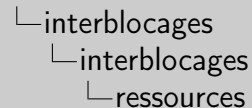
ressources

- ▶ une ressource est dite **non partageable** si elle doit être allouée de manière exclusive à un processus
- ▶ une ressource est dite **non préemptible** si elle ne peut être retirée à un processus sans dégâts
- ▶ Quid d'un graveur de CD ou d'une table traçante lors de l'exécution **alternée** de deux processus accédant à ces ressources
- ▶ l'exécution de tels processus risque de poser quelques difficultés.

ressources

Une demande de ressource **non partageable** et **non préemptible** attribuée à un autre, **bloque** un processus jusqu'à "disponibilité de la ressource"

graveur, scanneur, ...



Une demande de ressource **non partageable** et **non préemptible** attribuée à un autre, **bloque** un processus jusqu'à "disponibilité de la ressource"

graveur, scanneur, ...

un processus sera bloqué comme si il attendait la fin d'une lecture sur disque

interblocages

interblocages

dans **certaines situations** plusieurs processus se bloquent mutuellement indéfiniment

ce que l'on nomme **étreinte mortelle** ou **interblocage** ou **deadlock**

un exemple

Soit l'exemple d'une utilisation **simultanée** de deux ressources **non partageables** et **non préemptibles** comme le scanneur et le graveur de CD, par deux processus :

processus A :

- obtenir scanneur
- obtenir graveur
- scanner et graver
- rendre graveur
- rendre scanneur

processus B :

- obtenir graveur
- obtenir scanneur
- scanner et graver
- rendre scanneur
- rendre graveur

un exemple

une séquence possible d'exécution :

- 1 le processus A obtient le scanneur
- 2 le processus B obtient le graveur
- 3 le processus A demande le graveur et ne peut l'obtenir il est donc bloqué. Il possède déjà le scanneur
- 4 le processus B demande le scanneur et est donc bloqué. Il possède déjà le graveur.

Nous avons ainsi créé une situation d'interblocage. Les processus intéressés sont bloqués **de manière irréversible** (interblocage).

Nous pouvons illustrer cela par un **graphe cyclique**.

conditions nécessaires non suffisantes

Une telle situation ne peut se présenter que dans le cas suivant :

- ▶ ressources **non partageables**
- ▶ ressources **non préemptibles**
- ▶ **plusieurs** de ces mêmes ressources sont nécessaires **simultanément** à **plus d'un** processus

il s'agit de conditions **nécessaires non suffisantes**

└ interblocages

└ interblocages

└ conditions nécessaires non suffisantes

Une telle situation ne peut se présenter que dans le cas suivant :

- ressources **non partageables**
- ressources **non préemptibles**
- **plusieurs** de ces mêmes ressources sont nécessaires **simultanément** à **plus d'un** processus

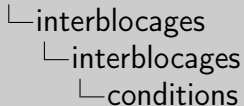
il s'agit de conditions **nécessaires non suffisantes**

- nécessaire = sans elle l'interblocage ne pourra survenir.
- non suffisante = n'implique pas qu'un interblocage surviendra

conditions

conditions nécessaires :

- ▶ une ressource **partageable** ou **préemptible** peut toujours être obtenue
- ▶ dans le cas d'utilisation d'une **ressource unique** le blocage est temporaire : le processus qui a la ressource se termine et cède la ressource à l'autre.



conditions nécessaires :

- une ressource **partageable** ou **préemptible** peut toujours être obtenue
- dans le cas d'utilisation d'une **ressource unique** le blocage est temporaire : le processus qui a la ressource se termine et cède la ressource à l'autre.

toutefois ces conditions réunies ne sont pas suffisantes pour que l'interblocage se présente. En effet selon la séquence d'ordonnancement choisie l'interblocage peut être évité

interblocages

interblocages - solutions ?

- ▶ ignorer (politique de l'autruche)
- ▶ ou bien
 - détecter et reprendre (graphes)
 - éviter
 - prévenir

ces techniques ne sont **pas sans coût**

solutions

ignorer

certain OS comme linux et windows ignorent les situations d'interblocage

est-ce suffisamment fréquent pour justifier la mise en oeuvre d'une gestion complexe ?

solutions

détecter-reprendre

► détecter

- on représente l'état de l'allocation/demande par un graphe
- on détecte les cycles dans le graphe

► reprendre

- suppression de processus
- préemption sur la ressource
- technique de Rollback (point de reprise)

- **détecter**
 - on représente l'état de l'allocation/demande par un graphe.
 - on détecte les cycles dans le graphe
- **reprendre**
 - suppression de processus
 - préemption sur la ressource
 - technique de Rollback (point de reprise)

Si nous représentons par $R1 \rightarrow P1$ le fait que la ressource $R1$ est allouée à $P1$
par $P2 \rightarrow R2$ le fait que $P2$ nécessite la ressource $R2$
cette situation peut être détectée par la présence d'un cycle
dans le graphe correspondant à l'état des ressources
(demandées/obtenues)

solutions

éviter

Attributions réfléchies \rightarrow nécessite des informations sur les **attributions futures**

Basé sur la notion d'**état sûr** :

état tel qu'il existe une séquence d'ordonnancement permettant à tous les processus présents de se terminer

2021-01-31

SYS2

└ interblocages
└ interblocages
└ solutions

solutions

éviter

Attributions réfléchies → nécessite des informations sur les **attributions futures**

Basé sur la notion d'**état sûr** :

état tel qu'il existe une séquence d'ordonnement permettant à tous les processus présents de se terminer

voir exemple tannenbaum

solutions

prévenir

- ▶ rendre partageable - cas du **spooler** et de son **répertoire**
- ▶ **allocation en bloc**- non optimal
- ▶ **ordonner** - ne fonctionne pas toujours (grand nombre de ressources)

solutions

spooler

Le problème de l'accès à l'imprimante est résolu par une **technique** de prévention : le spool

aucun processus n'accède directement à l'imprimante à l'exception du **spooler**
d'impression

solutions

spooler : **processus** dédié aux impressions

Les processus qui demandent des impressions écrivent dans des **fichiers temporaires** déposés dans un **répertoire dédié** au spooler

Ce dernier commande l'impression d'un fichier à la fois et évite ainsi le mélange des sorties.

Le spooler n'imprimera un fichier que si il est complet (fermé)

questions

- 1 un interblocage ralentit le processeur[V-F]
- 2 un interblocage **peut ne pas** être détecté par l'OS[V-F]
- 3 une ressource préemptible et non partageable peut occasionner un interblocage[V-F]

Questions ?



shutterstock · 104523269

remerciements

remerciements à P.Bettens et M.Codutti
pour la mise en page :-) Mba

Crédits

Ces slides sont le support pour la présentation orale de l'activité d'apprentissage **SYS2** à la HE2B-ÉSI

Crédits Crédits

La distribution opensuse
du système d'exploitation **GNU Linux**.

LaTeX/Beamer comme système d'édition.

GNU make, rubber, pdfnup, ... pour les petites tâches.

Images et icônes

deviantart, flickr, The Noun Project 