NSI · TERMINALE / COURS

Les processus et l'ordonnancement

Astrolab Cours · astrolabcours.fr

01 Le problème

Sur un ordinateur, la musique joue, le navigateur charge une page, un antivirus tourne en fond. Pourtant, un cœur de processeur ne fait qu'une chose à la fois. Comment tout avance en même temps ?

01 L'idée : le cuisinier qui jongle

Un cuisinier seul prépare cinq plats. Il met l'eau à chauffer, et pendant qu'elle chauffe, il coupe les légumes. Il passe d'un plat à l'autre quand l'un doit attendre. Chaque plat avance, et vu de la salle, tout semble se faire en même temps. Le cuisinier, c'est le processeur ; celui qui décide de l'ordre, c'est l'**ordonnanceur**.

01 Programme et processus

> Un **programme** est un fichier d'instructions. Un **processus** est un programme **en cours d'exécution**, avec sa mémoire, ses fichiers ouverts et son état. Lancer deux fois le même programme crée deux processus.

Chaque processus a un numéro, le **PID**. Sous Linux, `ps` les liste et `kill 4127` arrête le processus 4127.

01 Les états d'un processus

État	Signification
élu	il s'exécute sur le processeur
prêt	il attend son tour
bloqué	il attend une ressource (lecture disque, clavier, réseau)

Un processus bloqué redevient prêt quand la ressource arrive ; il ne repasse pas directement à l'état élu.

L'ordonnancement

Le système d'exploitation choisit quel processus prêt devient élu.

- **Premier arrivé, premier servi** : simple, mais un long processus fait attendre tous les autres.
- **Plus court d'abord** : réduit l'attente moyenne, mais il faut connaître les durées.
- **Tourniquet** (*round robin*) : chacun s'exécute un court **quantum** de temps, puis repasse en fin de file. C'est la base des systèmes actuels : personne n'attend trop longtemps.

L'interblocage

Le processus P1 détient l'imprimante et attend le scanner ; P2 détient le scanner et attend l'imprimante.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/processus/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits

Aucun ne lâche ce qu'il a : ils attendent pour toujours. C'est un **interblocage** (*deadlock*).

Exemple guidé

Trois processus arrivent en même temps, dans l'ordre P1, P2, P3. Ils ont besoin de 3, 5 et 2 unités de temps. Compare les dates de fin selon l'algorithme d'ordonnancement.

Algorithme	Ordre d'exécution	Fins (P1, P2, P3)
Premier arrivé	P1, P2, P3	3, 8, 10
Plus court d'abord	P3, P1, P2	5, 10, 2
Tourniquet (quantum 2)	P1, P2, P3, P1, P2	7, 10, 6

1. **Premier arrivé** : chacun passe à son tour, jusqu'au bout. Moyenne des fins : $(3 + 8 + 10)/3 \approx 6{,}3$.
2. **Plus court d'abord** : P3 passe en premier. Moyenne : $(5 + 10 + 2)/3 \approx 5{,}7$, la meilleure ici.
3. **Tourniquet** : chacun reçoit 2 unités puis retourne en fin de file. P1 fait 0 à 2, P2 de 2 à 4, P3 de 4 à 6 (terminé), P1 de 6 à 7 (terminé), P2 de 7 à 10. Moyenne $\approx 7{,}7$, mais aucun processus n'attend longtemps avant de démarrer.

Retour au problème

L'ordonnanceur découpe le temps en tranches très courtes et passe d'un processus à l'autre des centaines de fois par seconde. La musique, le navigateur et l'antivirus avancent chacun un peu, à tour de rôle, assez vite pour qu'on croie à la simultanéité.

Les pièges

- Confondre programme (le fichier) et processus (l'exécution).
- Un processus bloqué passe par l'état prêt avant d'être élu.
- Plusieurs processus « en même temps » sur un cœur : ils alternent, ils ne sont pas simultanés.

© Astrolab Cours · astrolabcours.fr/nsi/terminale/processus/ · Tous droits réservés : usage personnel uniquement, republication et usage commercial interdits