



III PARALLÉLISME, MÉCANISMES DE BASE

Tâches & types protégés



12 - PRINCIPES DE LA CONCURRENCE

Programmation Concurrente - LI330
Université P. & M. Curie - année scolaire 2013/2014

PrC



SÉQUENTIEL VERSUS PARALLÈLE

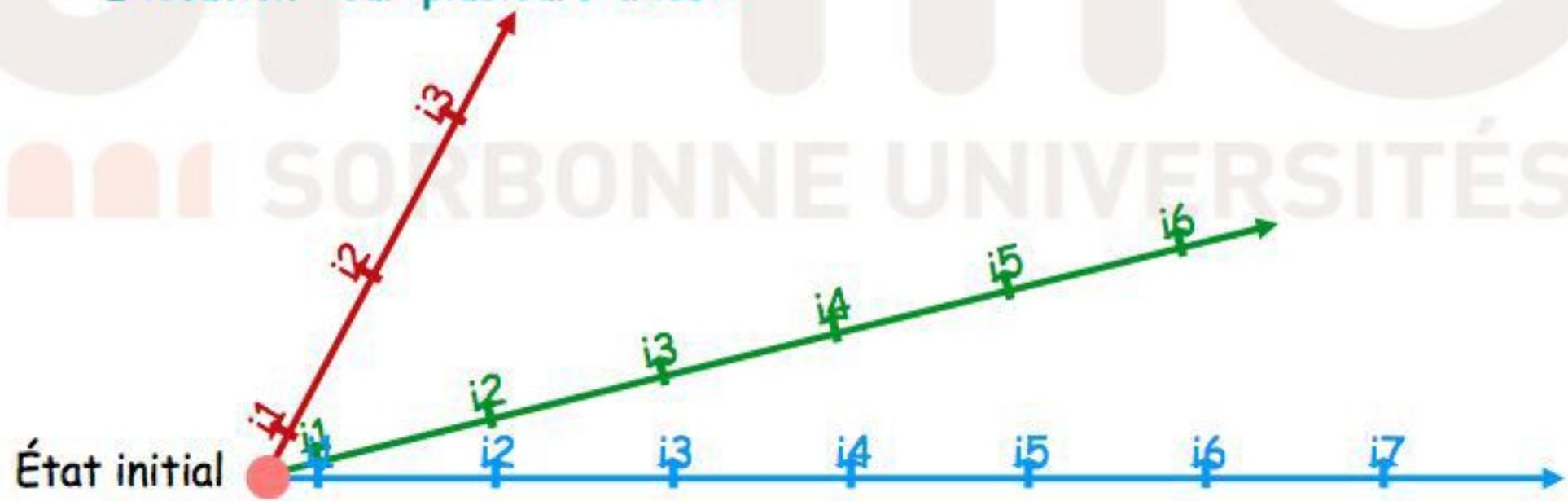
Séquentiel

- Un seul flux d'exécution
 - Exécution «sur un seul axe»



Parallèle

- Plusieurs flux d'exécution
 - Exécution «sur plusieurs axes»



SORBONNE UNIVERSITÉS

● Application séquentielle

- Le compteur ordinal
- La valeur des registres
- Les segments de mémoire ouverts

● Applications parallèle

- Le compteur ordinal
- La valeur des registres
- Les segments de mémoire ouverts

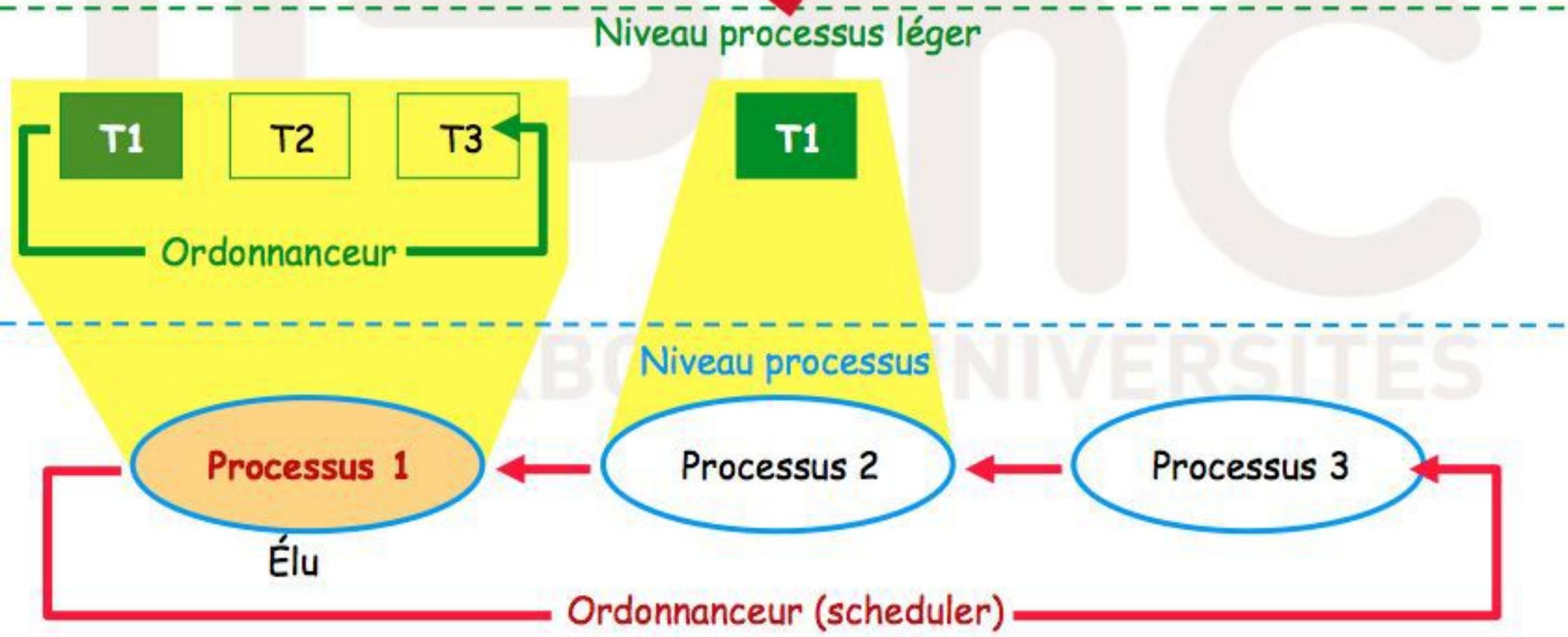
} pour chaque flux d'exécution

- Un seul segment partagé => thread (processus légers)
- Un segment par flux d'exécution => multi-processus
 - Sur un seul processeur
 - Sur plusieurs processeurs
- Attention: rien n'empêche de mélanger processus et processus légers dans une application



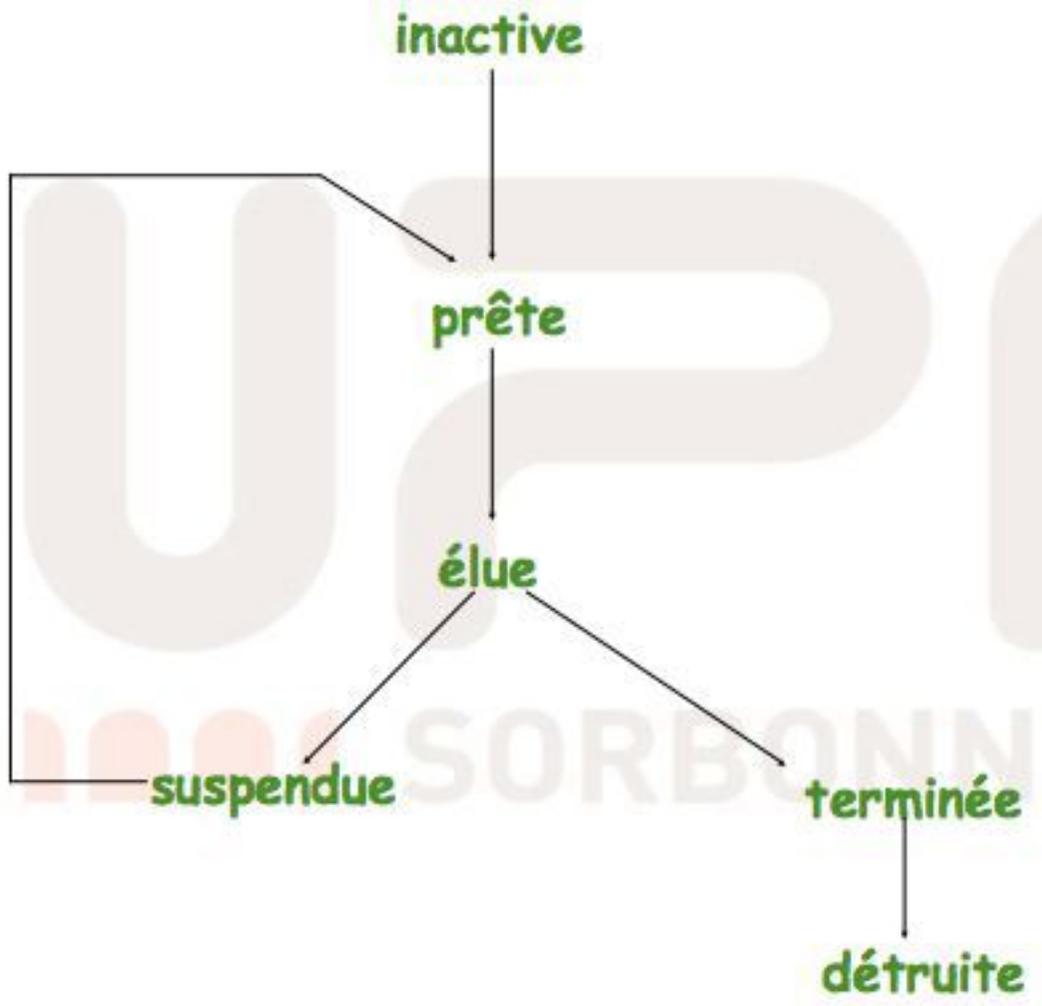
PROCESSUS ET PROCESSUS LÉGERS EN TEMPS PARTAGÉ

☒ Un espace mémoire autonome inaccessible par d'autres flux d'exécution





CYCLE DE VIE (SIMPLIFIÉ) D'UNE TÂCHE



● Cycle similaire pour
● les processus
● les processus légers (threads)

SORBONNE UNIVERSITÉS

Mono-processeur

-  Pseudo-parallélisme

-  Temps partagé

 -  Commutation «aléatoire»

-  Logique d'entrelacement des instructions

 -  Possibilité de partager l'espace mémoire (processus légers)

Multi-processeur (mono ou multi-machines)

-  Parallélisme réel

-  Instructions exécutées simultanément

-  Logique d'exécution simultanée

 -  Impossibilité de partager un espace mémoire



UNE GRANDE VARIÉTÉ DE MACHINES PARALLÈLES

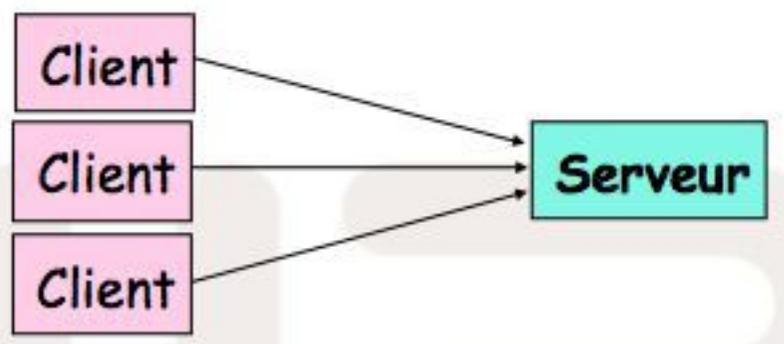
Historique, classification de Flynn (1972)

	SI	MI
SD	SISD (v. Neuman)	MISD (RISC...)
MD	SIMD (vectoriel)	MIMD (multi-processeur)

- Super-calculateurs (vectoriel)
- Shared everything, shared disks, shared nothing
 - Hybridation possible
- Multi-machines
 - Réseau local
 - Réseaux rapide (cluster)
 - Internet
- Besoin de paradigmes particuliers de programmation
 - Pas encore de solution pleinement satisfaisante mais «ça marche»

Il faut pouvoir écrire des programmes qui s'exécutent sur ces machines!

Modèle client-serveur



Modèle maître-esclave

