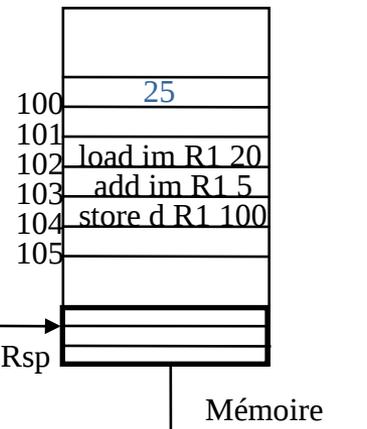
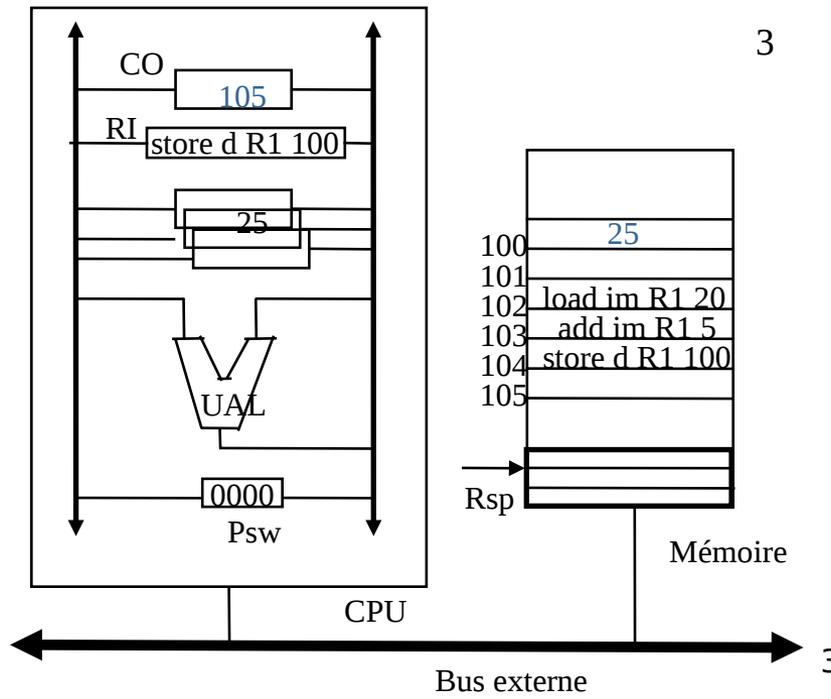
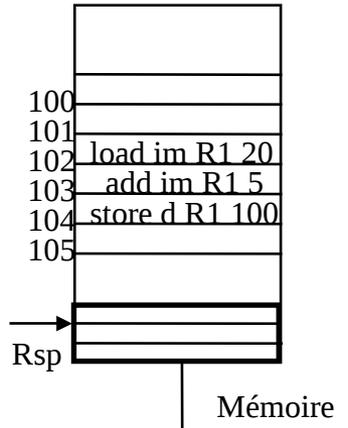
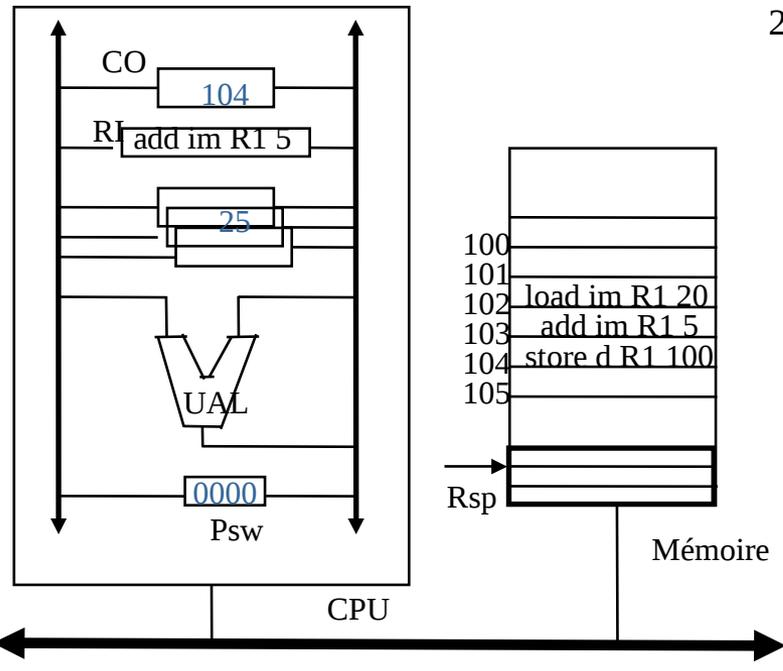
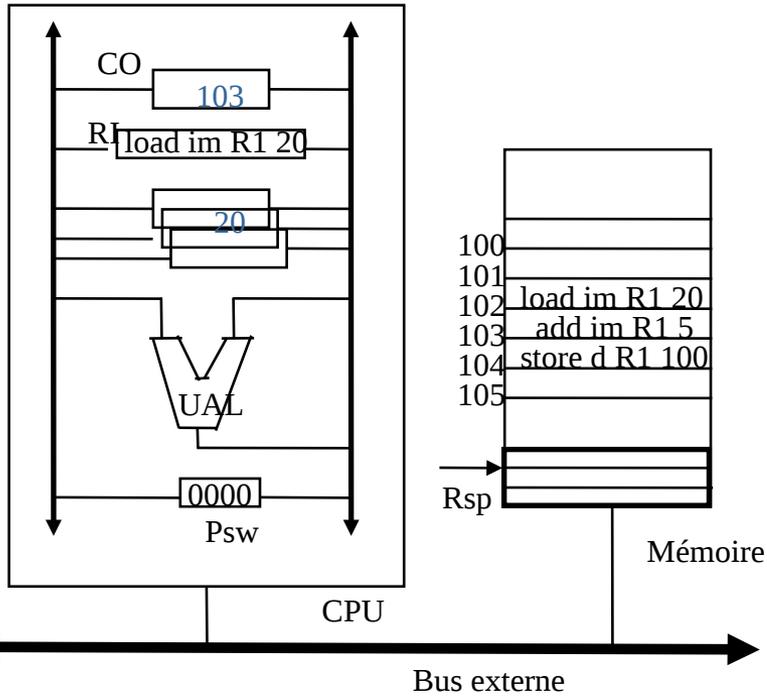


La gestion des exécutions des programmes

- ▣ Notion de processus
- ▣ Algorithmes d'ordonnancement
- ▣ L'exemple de Linux

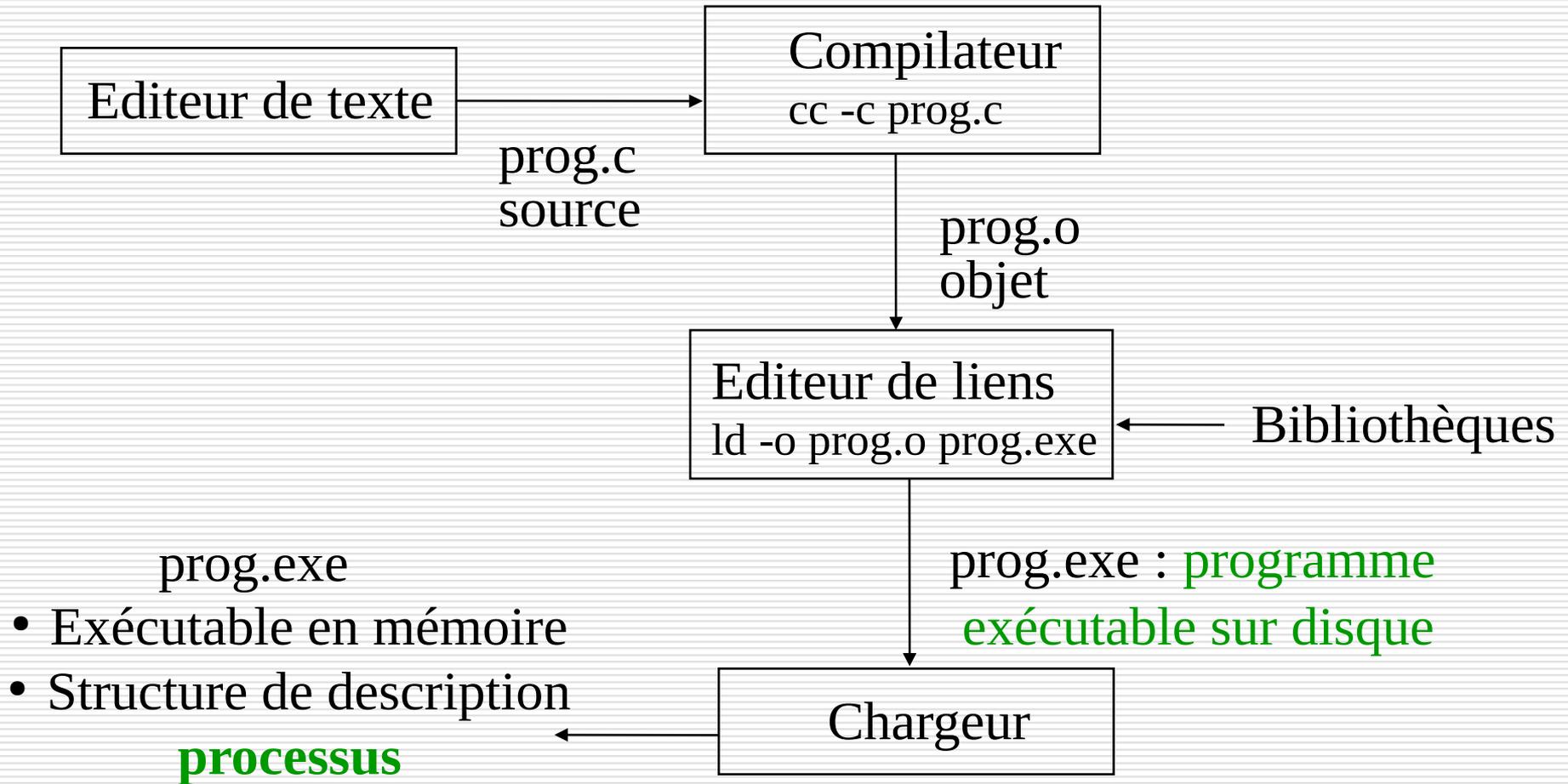
Notion de processus



- ✓ CO : compteur ordinal
- ✓ PSW : registre d'état
- ✓ RI : registre instruction

- Un **processus** est un **programme en cours d'exécution** auquel est associé un environnement processeur (CO, PSW, RSP, registres généraux) et un environnement mémoire appelés contexte du processus.
- Un **processus** est l'**instance dynamique d'un programme** et incarne le fil d'exécution de celui-ci dans un espace d'adressage protégé (objets propres : ensemble des instructions et données accessibles)
- Un **programme réentrant** est un programme pour lequel il peut exister **plusieurs processus** en même temps.

Du programme au processus

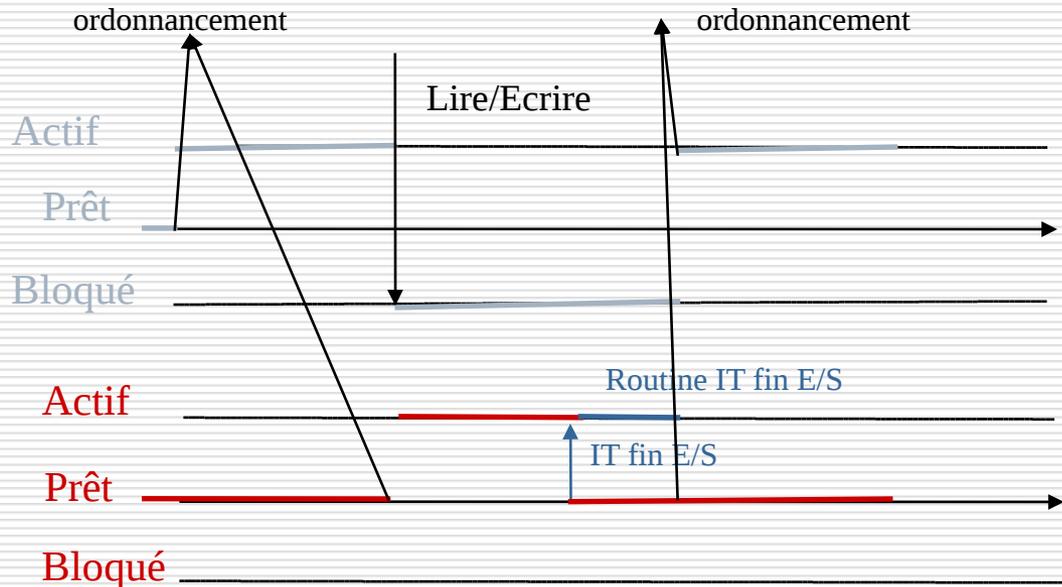
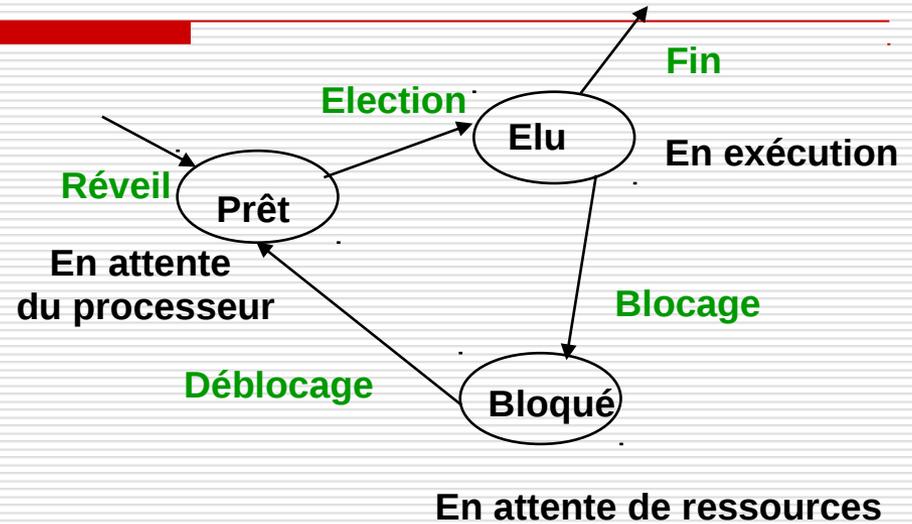


identificateur processus
état du processus
compteur instructions
contexte pour reprise (registres et pointeurs, piles,..)
pointeurs sur file d'attente et priorité(ordonnancement)
informations mémoire (limites et tables pages/segments)
informations de comptabilisation et sur les E/S, périphériques alloués, fichiers ouverts,..

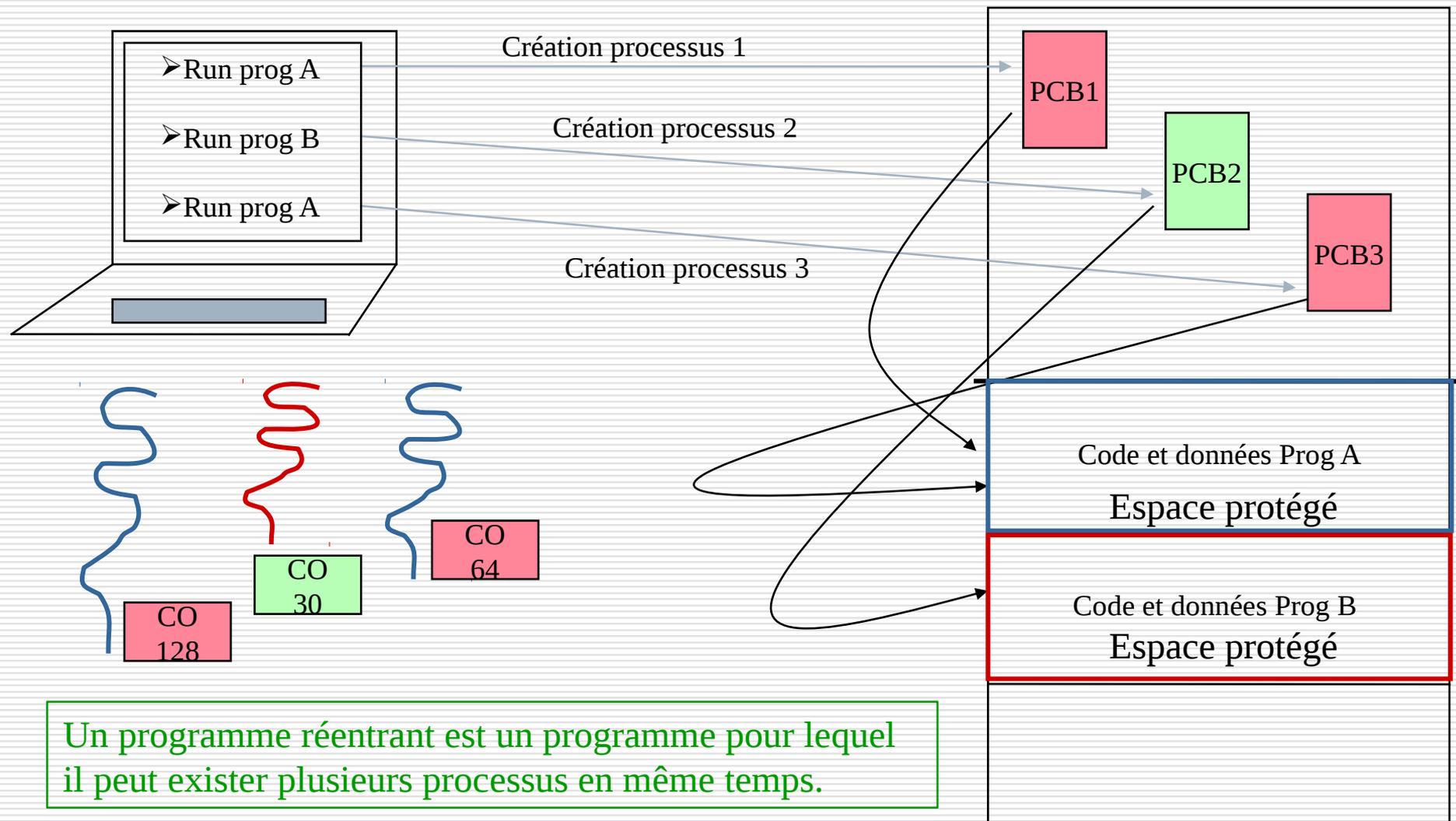
Bloc de contrôle de processus ou PCB

Systeme multiprocessus

Une ressource designe toute entite (materielle, logicielle) dont a besoin un processus pour s'executer (processeur, variable)

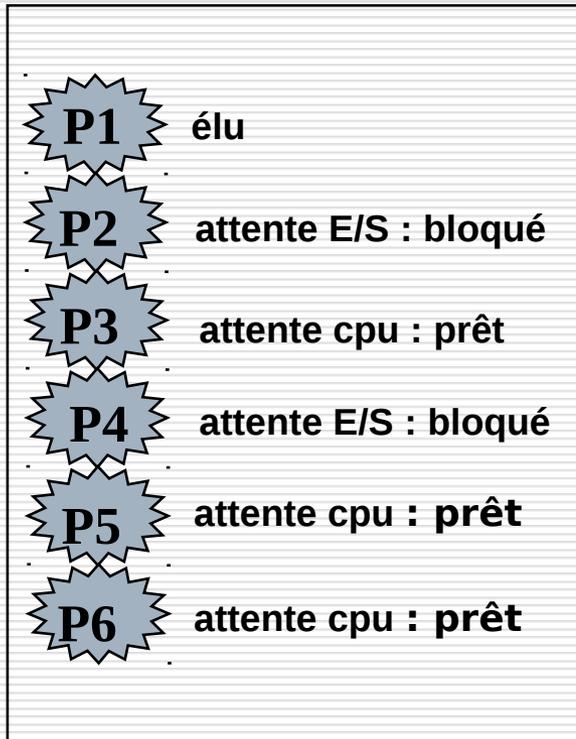


Programme réentrant



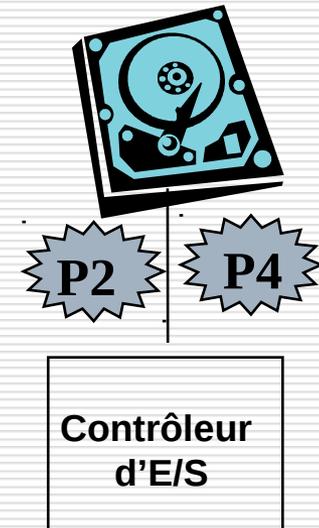
Politiques d'ordonnancement

Mémoire Centrale

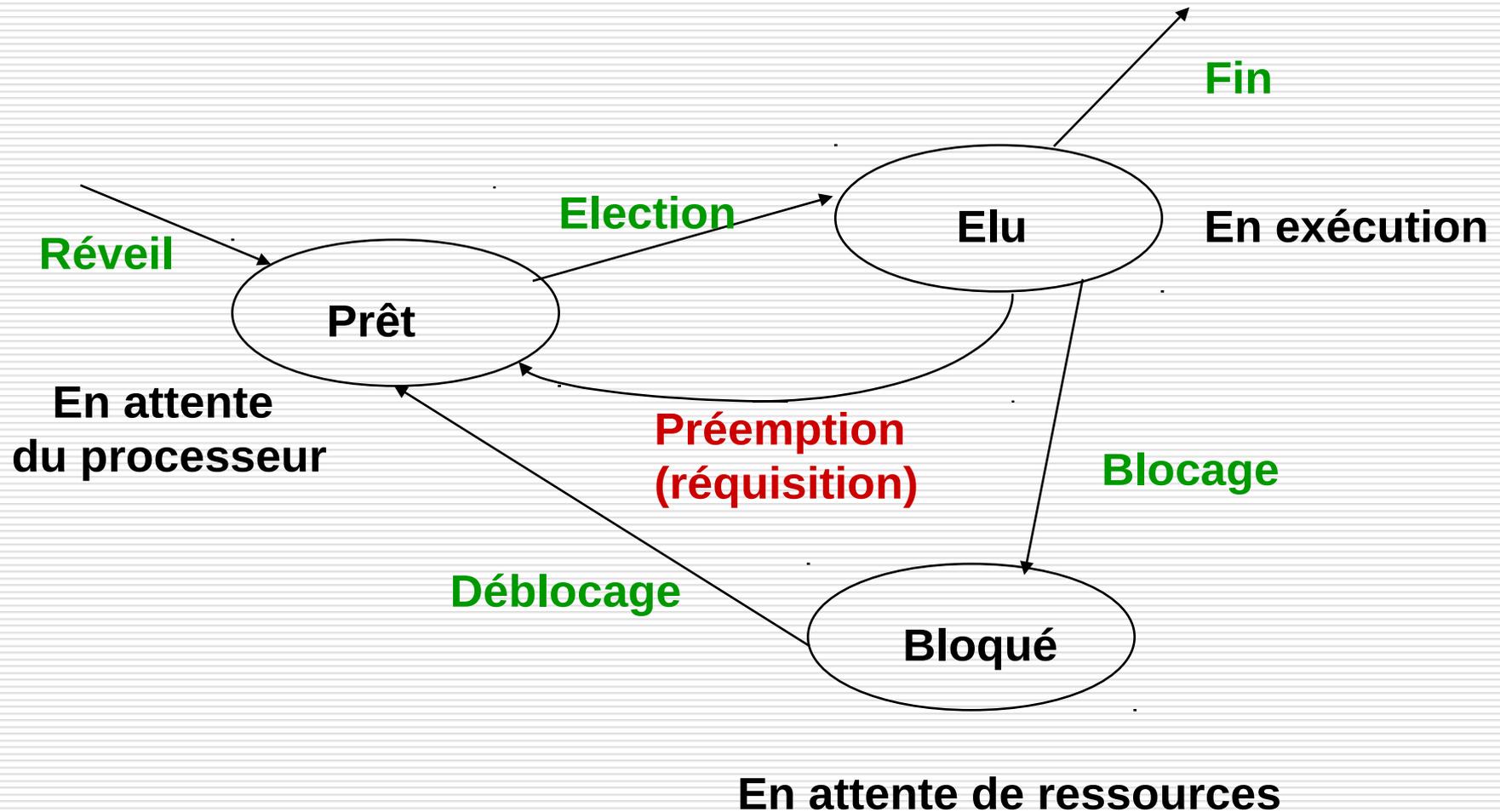


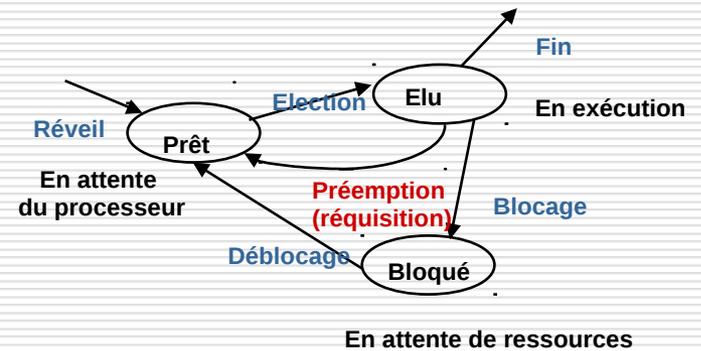
P1 se termine : quel processus exécuter
parmi les processus prêts P3, P5, P6 ?
P2 achève son entrées-sortie et devient prêt :
P1 doit-il poursuivre son exécution ou
laisser sa place à P2 (préemption) ?

Processeur



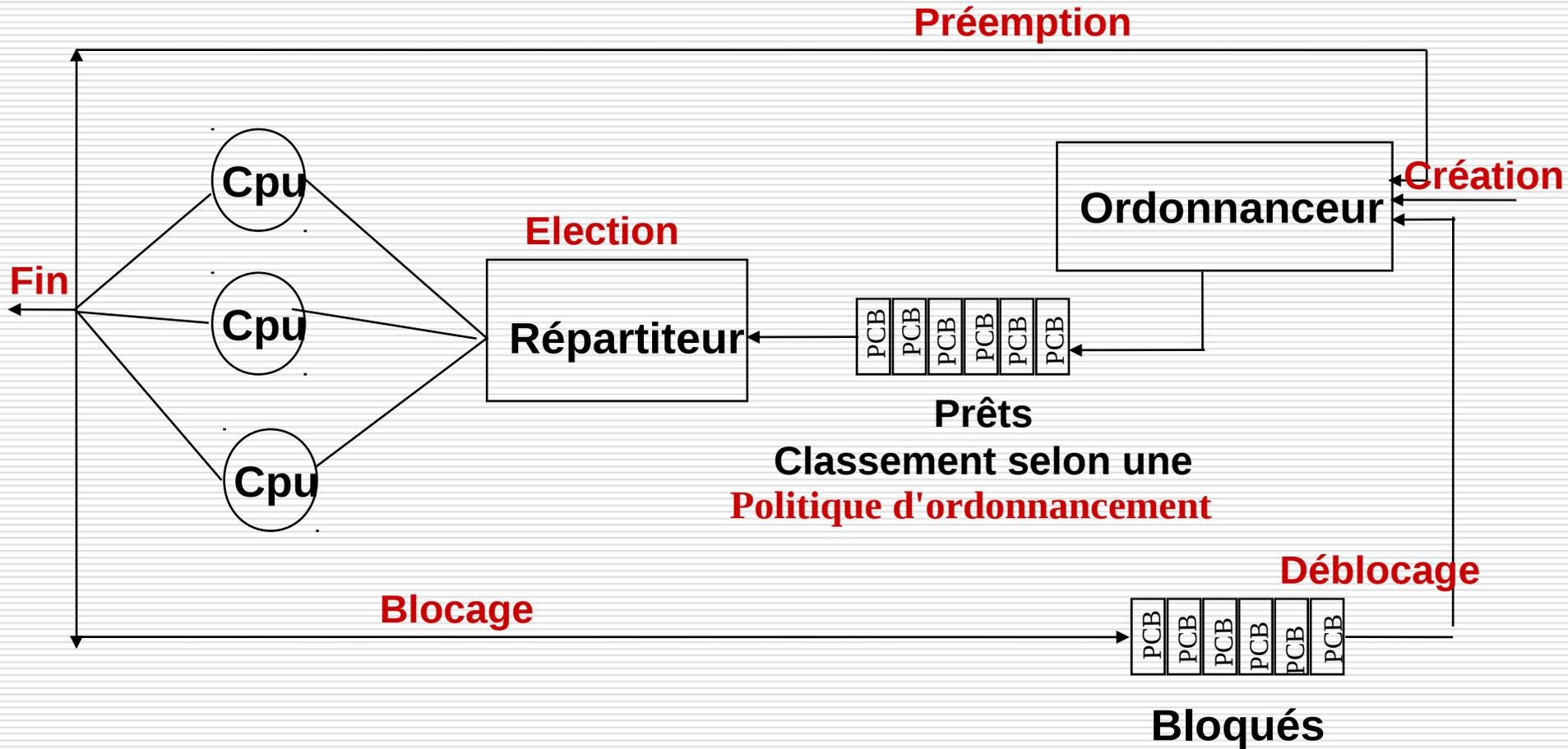
Bus





- **Election** : allocation du processeur
- **Préemption** : réquisition du processeur
 - ordonnancement non préemptif : un processus élu le demeure sauf s'il se bloque de lui-même
 - ordonnancement préemptif : un processus élu peut perdre le processeur
 - s'il se bloque de lui-même (état bloqué)
 - si le processeur est réquisitionné pour un autre processus (état prêt)

Ordonnanceur et répartiteur



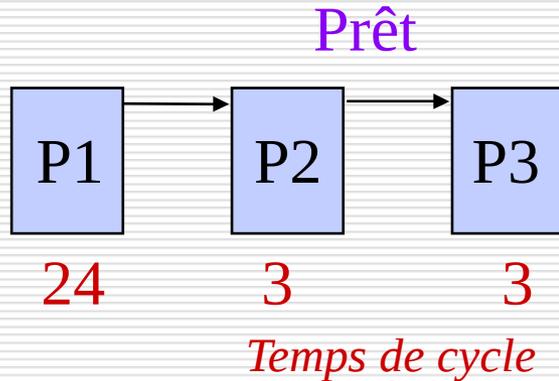
- Premier arrivé, premier servi
 - FIFO, sans réquisition

- Par priorités constantes

- Par tourniquet (round robin)

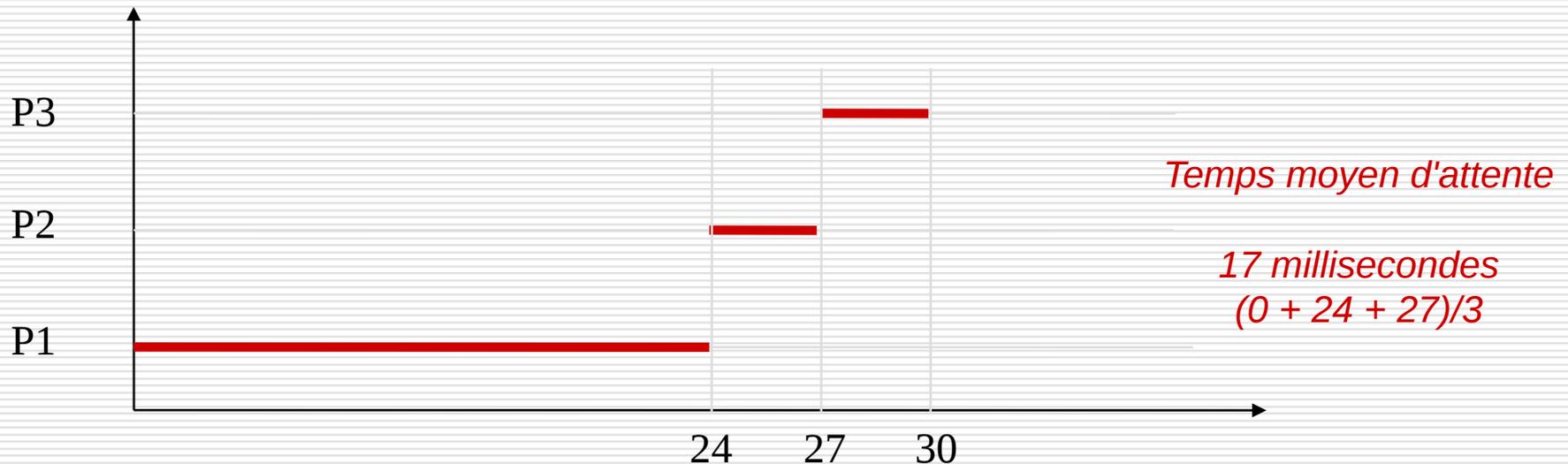
- Par files de priorités de priorités constantes multiniveaux avec ou sans extinction de priorité

FIFO, sans réquisition

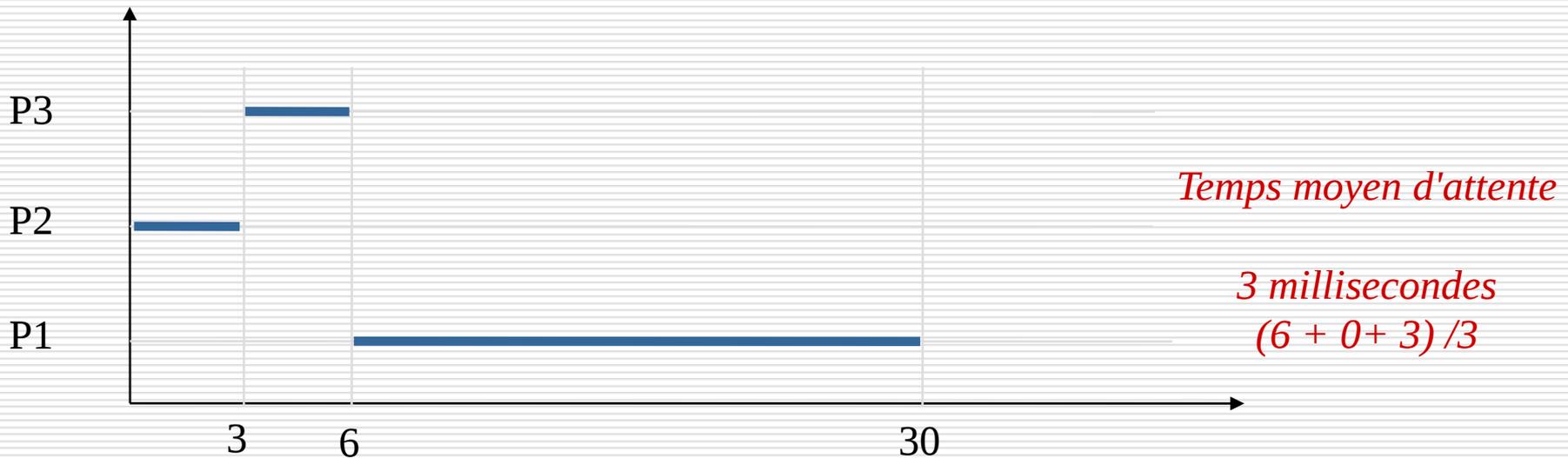
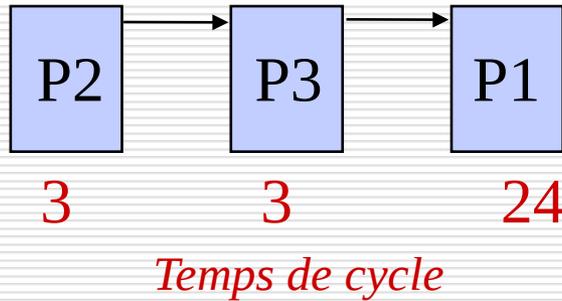


Tps d'attente = date début exécution – date soumission

Tps de réponse = date fin exécution – date soumission



Prêt



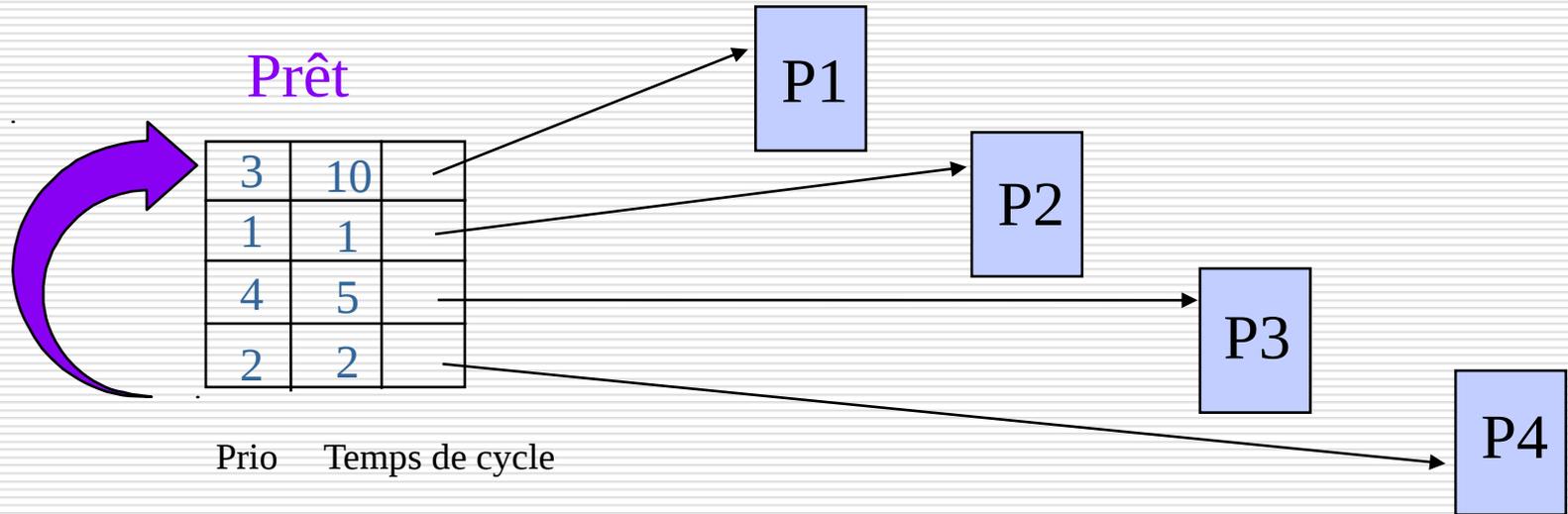
- Premier arrivé, premier servi

- Par priorités constantes
 - chaque processus reçoit une priorité
 - le processus de plus forte priorité est élu
 - Avec ou sans réquisition

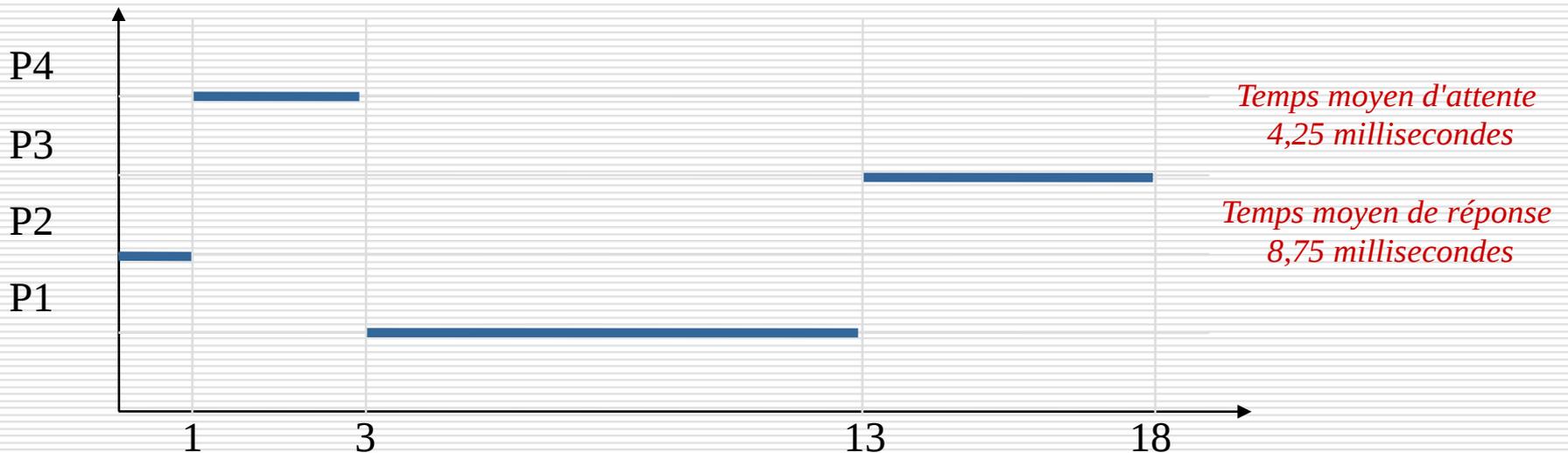
- Par tourniquet (round robin)

- Par files de priorités de priorités constantes multiniveaux avec ou sans extinction de priorité

Algorithme : avec priorités



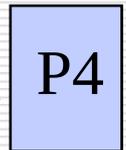
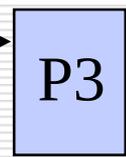
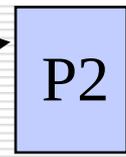
↳ Priorité : le plus petite valeur correspond à la plus forte priorité



Algorithme : avec priorités

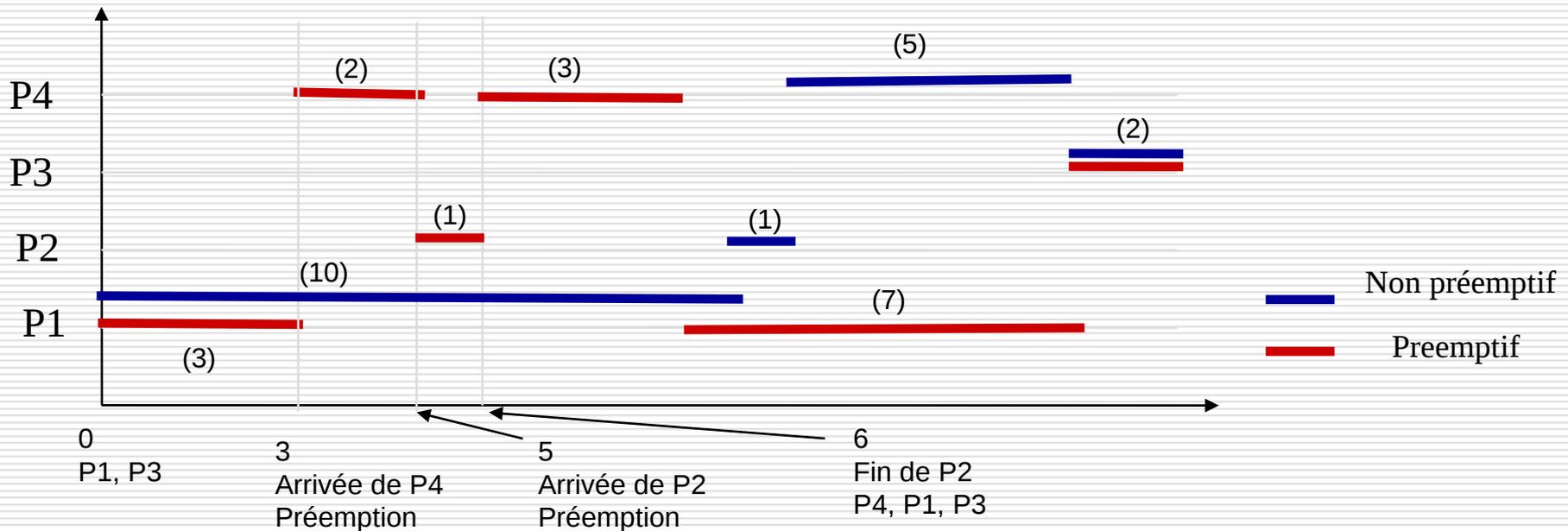
Prêt

0	3	10	
5	1	1	
0	4	2	
3	2	5	



Date de soumission Prio Temps de cycle

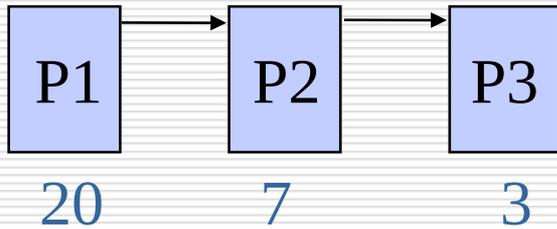
↳ Priorité : le plus petite valeur correspond à la plus forte priorité



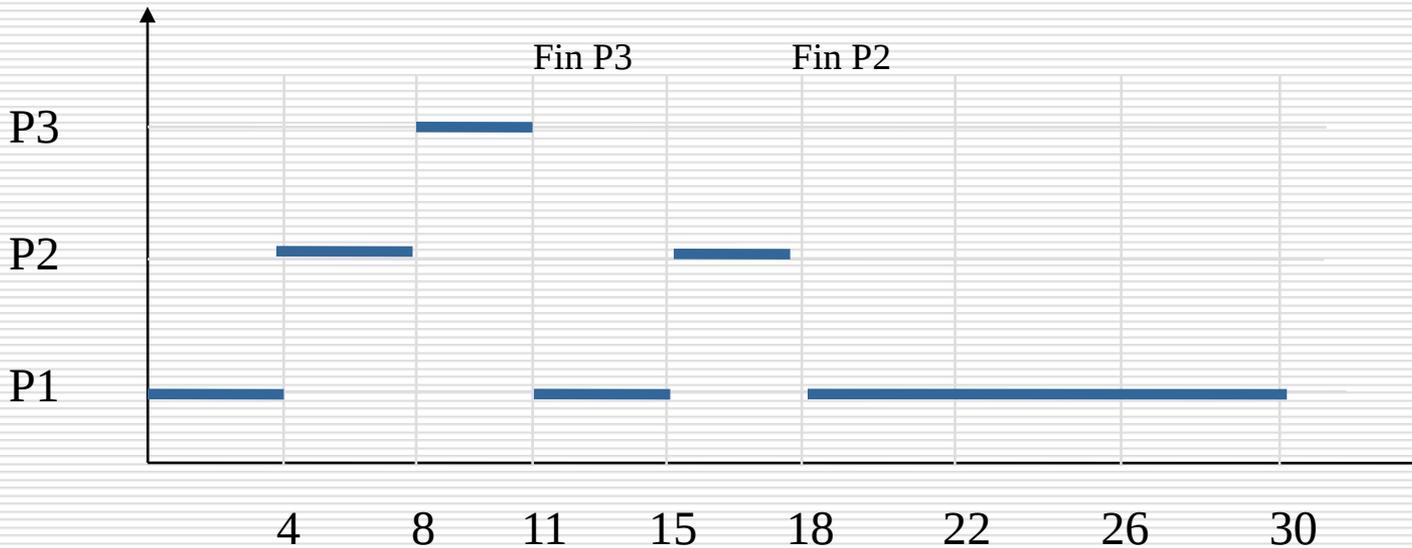
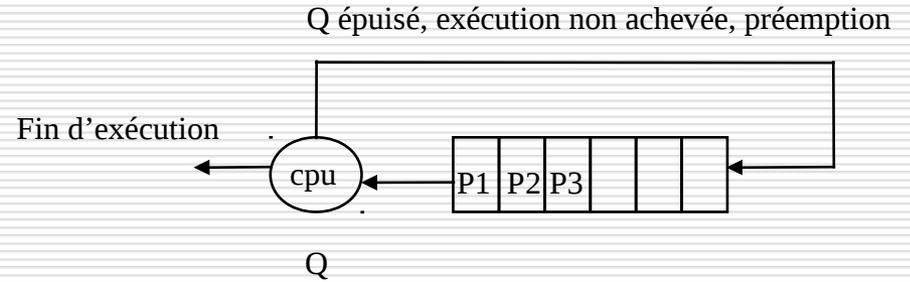
- Premier arrivé, premier servi
- Par priorités constantes
- Par tourniquet (round robin)
 - Définition d'un quantum = tranche de temps
 - Un processus élu s'exécute au plus durant un quantum; à la fin du quantum, préemption et réinsertion en fin de file d'attente des processus prêts
- Par files de priorités de priorités constantes multiniveaux avec ou sans extinction de priorité

Algorithme : tourniquet

Prêt



Temps de cycle



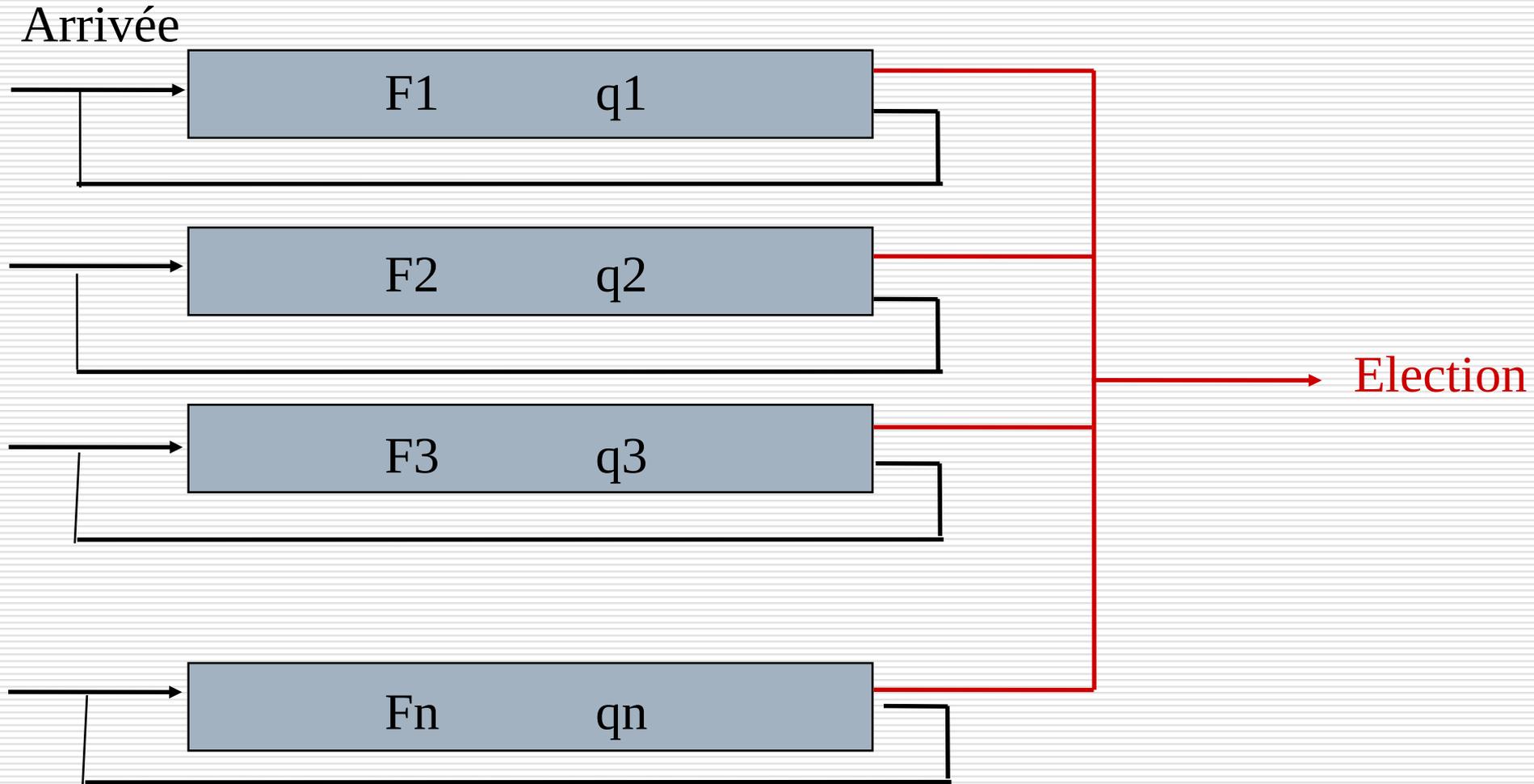
Quantum = 4

P1	P2	P3	P1	P2	P1
P2	P3	P1	P2	P1	
P3	P1	P2			

- Premier arrivé, premier servi
- Par priorités constantes
- Par tourniquet (round robin)

- Par files de priorités de priorités constantes multi-niveaux avec ou sans extinction de priorité
 - chaque file est associée à un quantum éventuellement différent
 - sans extinction : un processus garde toujours la même priorité
 - avec extinction : la priorité d'un processus décroît en fonction de son utilisation de la cpu

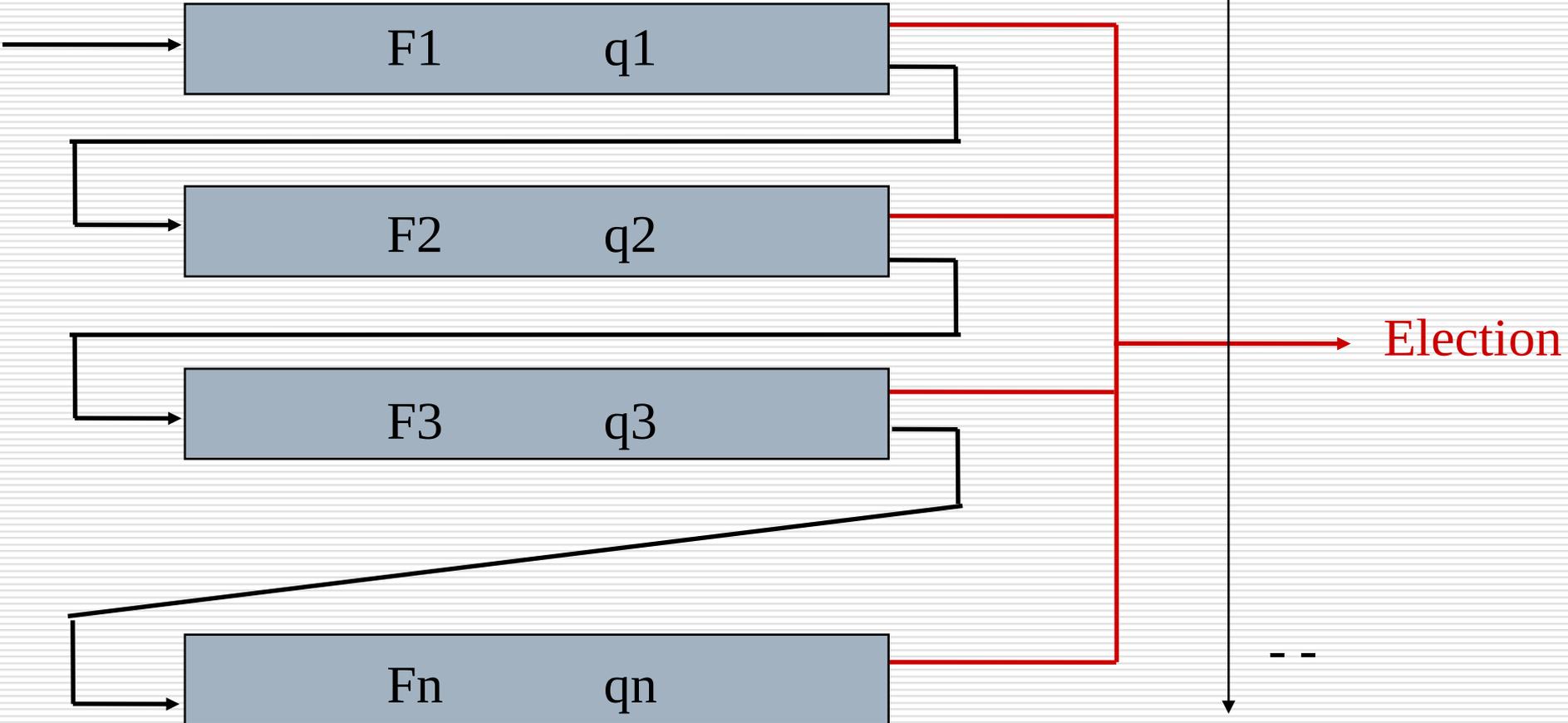
Prêt



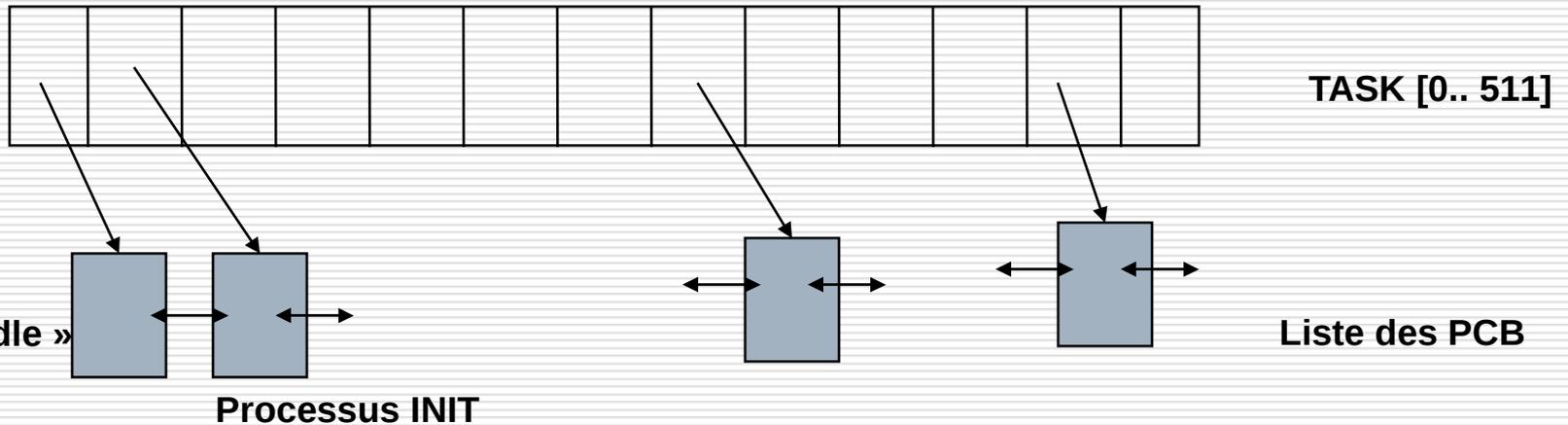
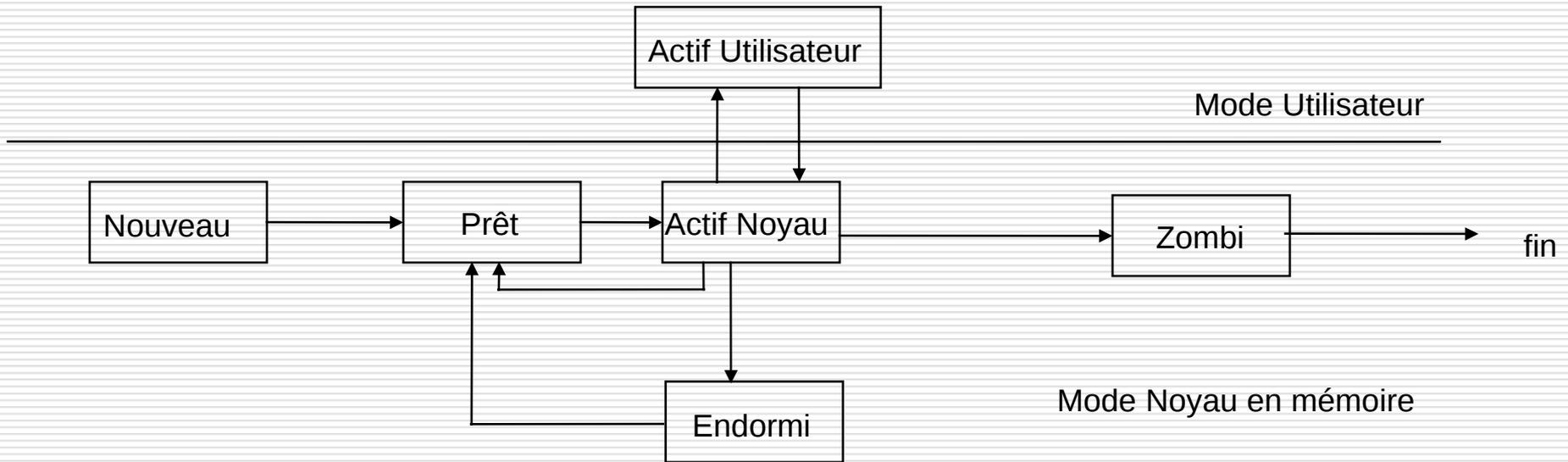
Prêt

Priorité

Arrivée

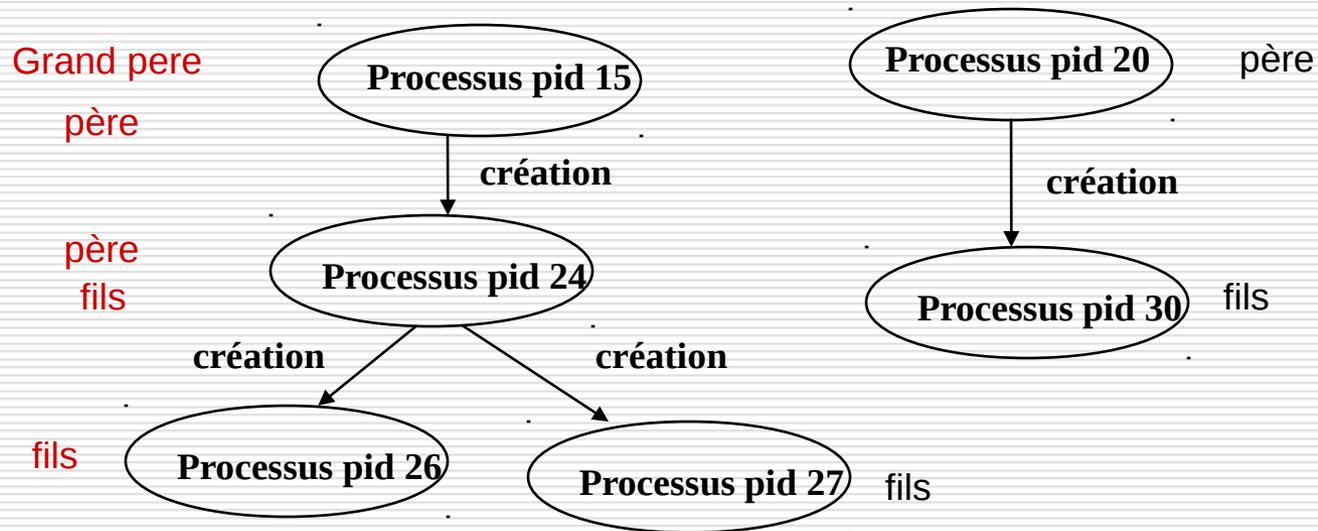


Exemple : Linux



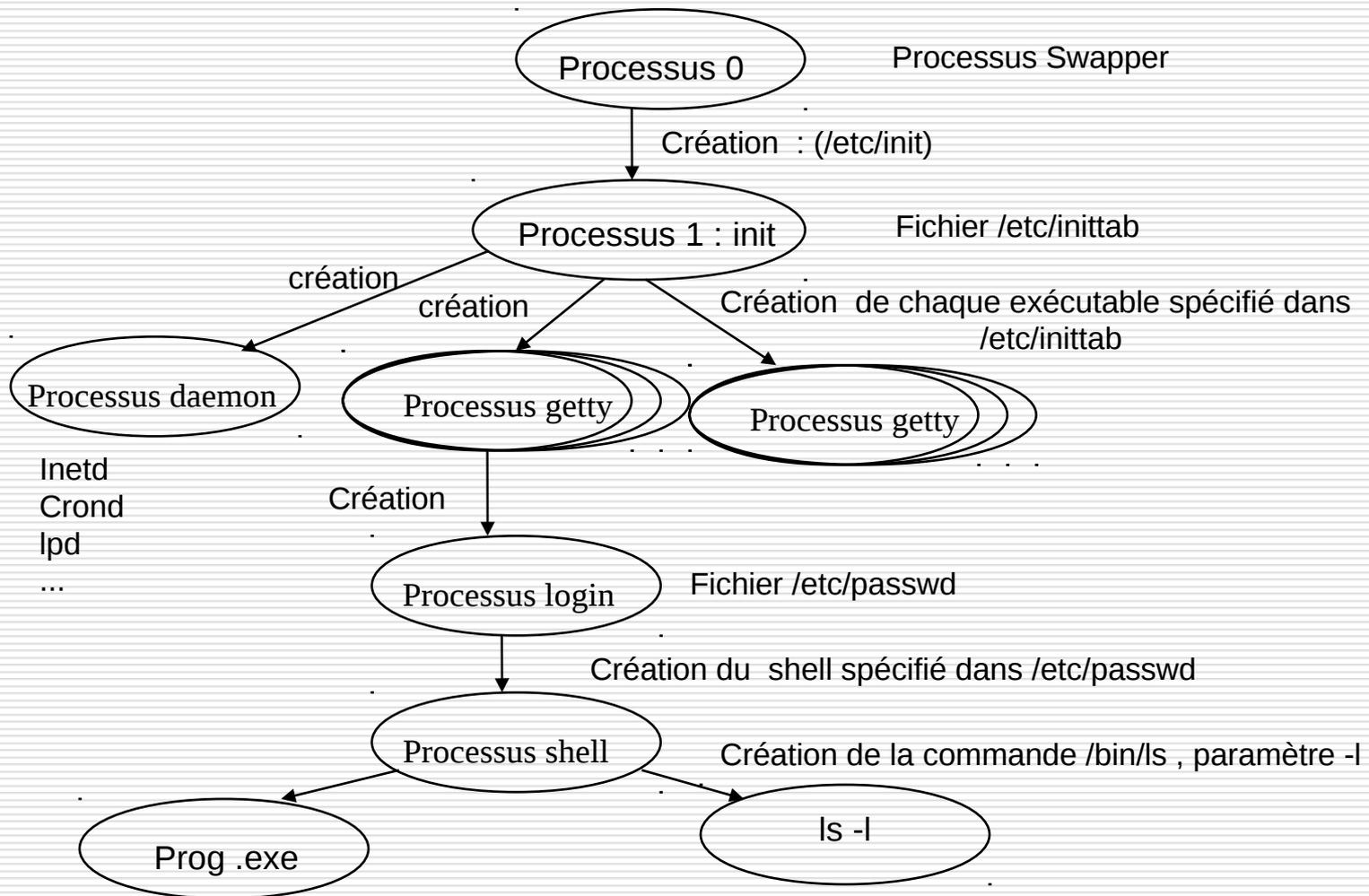
Tout processus Linux peut créer un autre processus Linux

→ Arborescence de processus avec un rapport père - fils entre processus créateur et processus créé



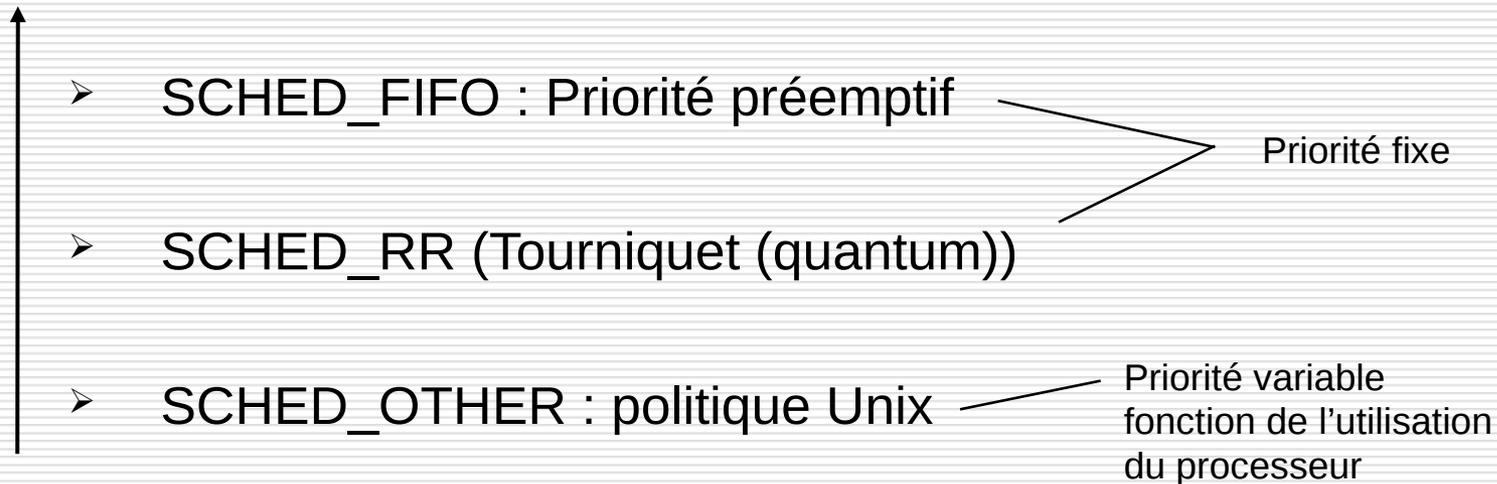
Processus Linux

Tout le système Linux repose sur ce concept arborescent



Trois classes d'ordonnancement (norme POSIX) :

Priorité supérieure



A l'instant t, le système élit (fonction GOODNESS du noyau)

- **Le processus SCHED_FIFO de plus forte priorité qui s'exécute jusqu'à sa fin ou jusqu'à préemption par un processus FIFO plus prioritaire**
- **Le processus SCHED_RR de plus forte priorité pour un quantum**
- **Le processus SCHED_OTHER de plus forte priorité**