

**M113**  
**Systèmes d'Exploitation et**  
**Programmation Système**

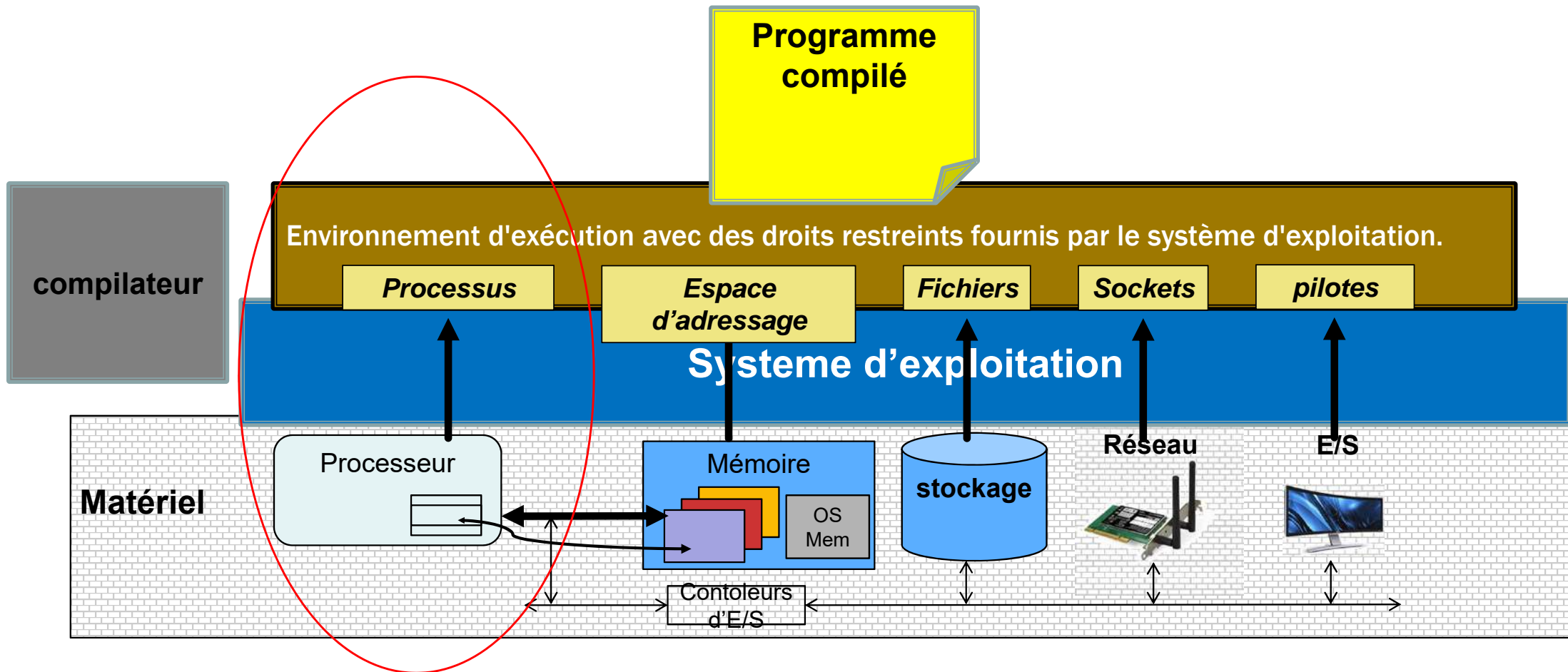
Prof. Fadoua YAKINE

**M113**  
**Systèmes d'Exploitation et**  
**Programmation Système**

Chapitre 2 : Gestion des processus

9 Octobre 2024  
Prof. Fadoua YAKINE

## Le SE est un ensemble de gestionnaires



## introduction

- Les processus sont l'une des **abstractions** les plus anciennes et les plus importantes pour représenter **l'exécution concurrente** de tâches au sein d'un système d'exploitation **multitâche**.
- Le GP est responsable de:
  - ◆ Allocation de la ressource processeur aux processus
  - ◆ Création, terminaison des processus
  - ◆ Suspension, reprise des processus
  - ◆ Synchronisation, communication inter-processus

## I ) Les Processus

## Le processeur

- Le "cerveau" de l'ordinateur est le processeur ou UC.
- Il extrait les **instructions** de la mémoire et les **exécute**.
- Le cycle de base de tout processeur consiste :
  - ◆ à extraire la première instruction de la mémoire, la décoder pour déterminer son type et ses opérandes, l'exécuter,
  - ◆ puis recommencer avec les instructions qui suivent→ C'est ainsi qu'un programme s'exécute.

## Définition d'un Processus:

- Un processus est une **abstraction**.
- Il représente un programme en cours d'exécution auquel est associé .
  - ◆ Environnement processeur (Compteur Ordinal, registres généraux...)
  - ◆ Environnement mémoire (zone de code, de données et de pile)} Contexte du processus
- Programme : entité statique.
- Processus : entité dynamique.
- **Concept central** de tout SE multiprogrammé (multitâche).

## Définition d'un Processus:





## États d'un processus

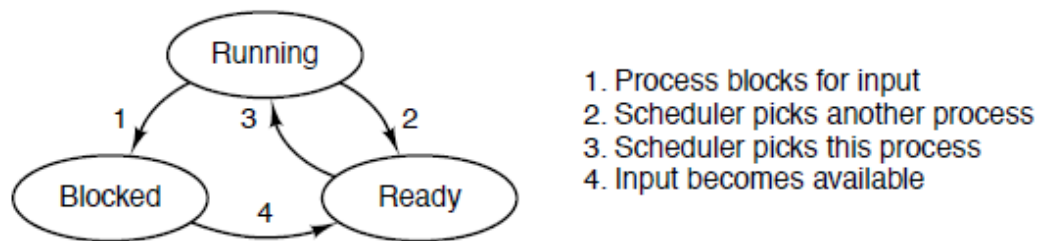


Figure 2-2. A process can be in running, blocked, or ready state. Transitions between these states are as shown.

### 4 Transitions :

1. Blocage
2. Préemption ou Réquisition
3. Election
4. Déblocage

1. **Elu** : lorsque le processus obtient le processeur et s'exécute (utilise le processeur en cet instant)
2. **Bloqué** : attente d'un événement externe ( ne pouvant pas s'exécuter tant qu'un événement externe ne se produit pas )
3. **Prêt** : l'état d'attente du processeur ( exécutable , temporairement arrêté pour laisser s'exécuter un autre processus)

## Implémentation des processus

- Pour implémenter le modèle de processus, le SE gère une table (un tableau de structures), appelée **table de processus (ou bloc de contrôle de processus PCB process control bloc)**.
- Chaque entrée correspond à processus particulier.
- Le contenu type d'une table de processus concerne :
  - ◆ La gestion du processus
  - ◆ La gestion mémoire
  - ◆ La gestion des fichiers

## Bloc de contrôle du processus (PCB)

- Le bloc de contrôle d'un processus contient les informations suivantes :
  - ◆ un identificateur unique du processus (**pid**);
  - ◆ l'état courant du processus (élu, prêt, bloqué) ;
  - ◆ le contexte processeur du processus : la valeur du CO, la valeur des autres registres du processeur;
  - ◆ le contexte mémoire : ce sont des informations mémoire qui permettent de trouver le code et les données du processus en mémoire centrale;
  - ◆ des informations liées à l'ordonnancement du processus;
  - ◆ des informations sur les ressources utilisées par le processus, tels que les fichiers ouverts, les outils de synchronisation utilisés, etc.

## Bloc de contrôle du processus (PCB)

Figure 2.4 • Quelques champs d'une entrée type de la table des processus.

Gestion du processus	Gestion de la mémoire	Gestion de fichier
Registres	Pointeur vers un segment de texte	Répertoire racine
Compteur ordinal	Pointeur vers un segment de données	Répertoire de travail
Mot d'état du programme	Pointeur vers un segment de la pile	Descripteurs de fichiers
Pointeur de la pile		ID utilisateur
État du processus		ID du groupe
Priorité		
Paramètres d'ordonnancement		
ID du processus		
Processus parent		
Groupe du processus		
Signaux		
Heure de début du processus		
Temps de traitement utilisé		
Temps de traitement du fils		
Heure de la prochaine alerte		

## II ) L'Ordonnancement

## Qu'est-ce qu'un algorithme d'ordonnancement ?

### 1 Définition

Un algorithme d'ordonnancement est un ensemble de règles et de procédures qui permettent de déterminer l'ordre d'exécution des processus dans un système informatique.

### 2 Objectifs

Les principaux objectifs sont de maximiser l'utilisation des ressources, de minimiser le temps d'attente et de garantir l'équité entre les processus.

### 3 Complexité

La conception d'algorithmes d'ordonnancement efficaces est un défi complexe, car de nombreux facteurs entrent en jeu.

## ORDONNANCEMENT des processus

- La fonction d'ordonnancement gère **le partage du processeur** entre les différents processus en attente pour s'exécuter.
- Elle s'intéresse plus particulièrement aux **transitions** pouvant exister entre l'état **prêt** et l'état **élu**.
- Le passage de l'état prêt vers l'état élu constitue l'opération d'**élection** : c'est l'allocation du processeur à **un** des processus prêts

## ORDONNANCEMENT des processus

- Lorsqu'un ordinateur est multiprogrammé, il a fréquemment plusieurs **processus** ou **threads** en concurrence pour le CPU en même temps.
- Cette situation se produit chaque fois que deux ou plusieurs d'entre eux sont simultanément à **l'état prêt**.
- La partie du système d'exploitation qui fait le choix s'appelle **Ordonnanceur (Scheduler )**
- L'ordonnanceur utilise un algorithme appelé **algorithme d'ordonnancement (scheduling algorithm)** ou **politique d'ordonnancement**.



## Ordonnancement préemptif et non préemptif

- L'ordonnancement s'intéresse plus particulièrement aux transitions pouvant exister entre l'état prêt et l'état élu .
- Le passage de l'état prêt vers l'état élu constitue l'opération d'élection : c'est l'allocation du processeur à un des processus prêts.
- Réquisition (**préemption**) du processeur: le processeur est retiré au processus élu alors que celui-ci dispose de toutes les ressources nécessaires à la poursuite de son exécution.

## Ordonnancement préemptif et non préemptif

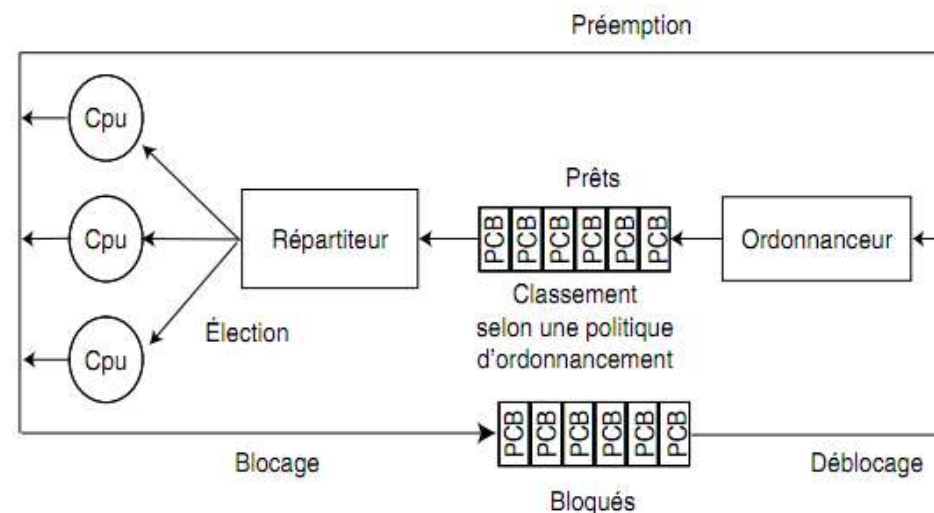
- Réquisition (**préemption**) du processeur: le processeur est **retiré** au processus élu alors que celui-ci dispose de toutes les ressources nécessaires à la poursuite de son exécution.
- **L'ordonnancement préemptif** (preemptive scheduling) s'oppose à la méthode d'**exécution jusqu'à achèvement** (**ordonnancement non préemptif**) des premiers systèmes à traitements par lots.
- Les algorithmes d'**ordonnancement non préemptif** sont simples et faciles à mettre en œuvre mais **inadaptés** à des systèmes généraux à plusieurs utilisateurs.

## Ordonnancement préemptif et non préemptif

- Si l'ordonnancement est **non préemptif**, la transition de l'état élu vers l'état prêt est **interdite** : un processus quitte le processeur s'il a terminé son exécution ou s'il se bloque;
- Si l'ordonnancement est **préemptif**, la transition de l'état élu vers l'état prêt est **autorisée** : un processus quitte le processeur s'il a terminé son exécution, s'il se bloque ou si le processeur est réquisitionné.

## Entités systèmes responsables de l'ordonnancement

- Les processus prêts et les processus bloqués sont gérés dans deux files d'attentes distinctes qui chaînent leur PCB.
- Le module **ordonnanceur** (scheduler) trie la file des processus prêts de telle sorte que le prochain processus à élire soit toujours en tête de file.
- Le **répartiteur** (dispatcher) alloue un processeur parmi les processeurs libres à la tête de file de la liste des processus prêts et réalise donc l'opération d'élection.



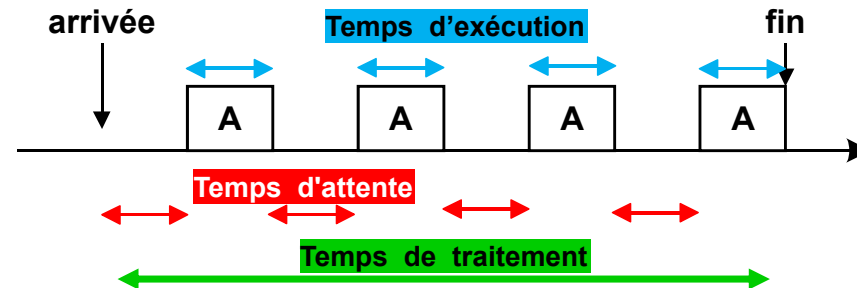
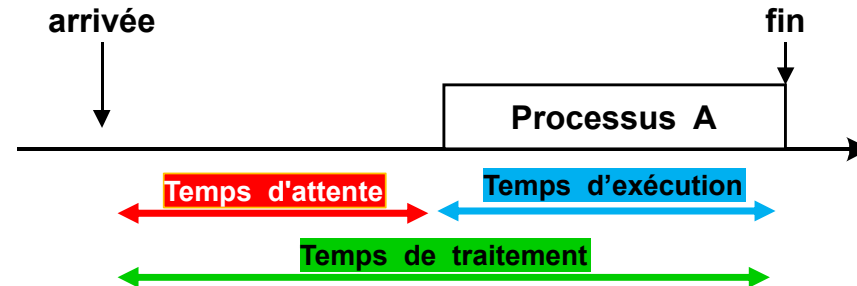
## Politiques d'ordonnancement

- La politique d'ordonnancement détermine quel sera le prochain processus élu.
- Selon si la préemption est autorisée ou non, la politique d'ordonnancement sera de type préemptive ou non.

## Politiques d'ordonnancement

- 1) Politique Premier Arrivé, Premier Servi (FCFS ou FIFO)
- 2) Politique Plus Court d'Abord (Shortest Job First, SJF)
- 3) Politique par priorité
- 4) Politique du tourniquet (round robin)
- 5) Politique du TEMPS RESTANT LE PLUS COURT (EN PREMIER) (SRT)

## TEMPS DE TRAITEMENT & TEMPS D'ATTENTE



**Temps de traitement (réponse)** = temps de fin - instant d'arrivée

**Temps d'attente** = temps de traitement - temps d'exécution

## Politique Premier Arrivé, Premier Servi FIFO - First-IN, First-OUT

### Principe

Les processus sont servis dans l'ordre de leur arrivée dans la file d'attente.

### Avantages

Simplicité de mise en œuvre

### Inconvénients

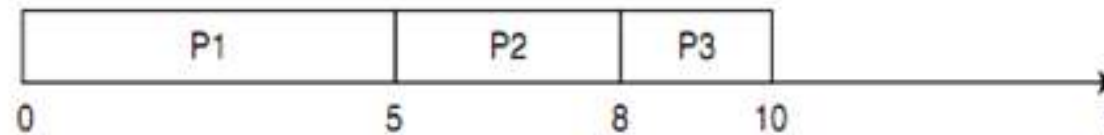
Manque d'optimisation et possibilité de famine (starvation) pour les processus longs.



## FIFO

Processus	durée	Ordre d'arrivée
P1	5	1
P2	3	2
P3	2	3

- Tracer le diagramme de *GANTT*



- Donner le temps de traitement moyen

$$T_{\text{traitement}} = (5 + 8 + 10) / 3 = 7,67ut$$

- Donner le temps d'attente moyen

$$T_{\text{attente}} = (0 + 5 + 8) / 3 = 4,33ut$$

## Politique Plus Court d'Abord (Shortest Job First)

### Principe

l'ordre d'exécution des processus est fonction de leur temps d'exécution.

Le processus de plus petit temps d'exécution est celui qui est ordonnancé en premier.

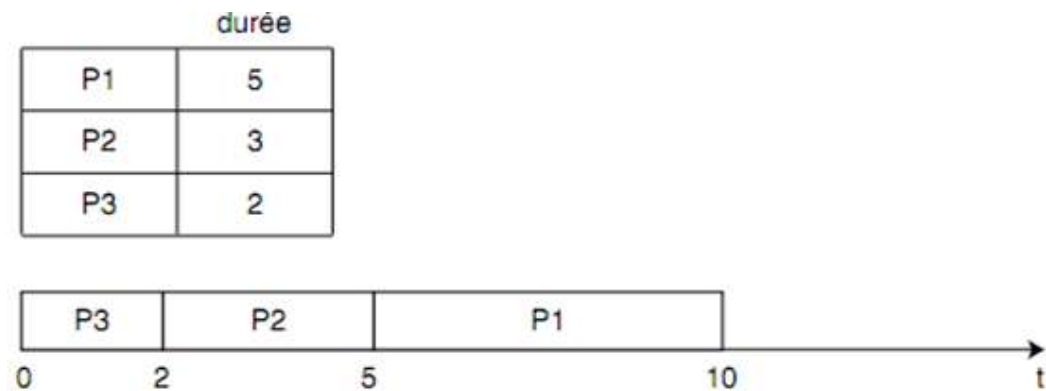
### Avantages

Minimise le temps d'attente moyen et favorise les processus courts.

### Inconvénients

famine : pénaliser les processus de grand temps d'exécution

## Politique Plus Court d'Abord (Shortest Job First)



temps de réponse moyen =  $(2 + 5 + 10) / 3 = 5,67$

temps d'attente moyen =  $(0 + 2 + 5) / 3 = 2,33$

Politique « Plus Court d'Abord ».

## Politique par priorité

### Principe

Chaque processus possède une priorité.

À un instant donné, le processus élu est le processus prêt de plus forte priorité.

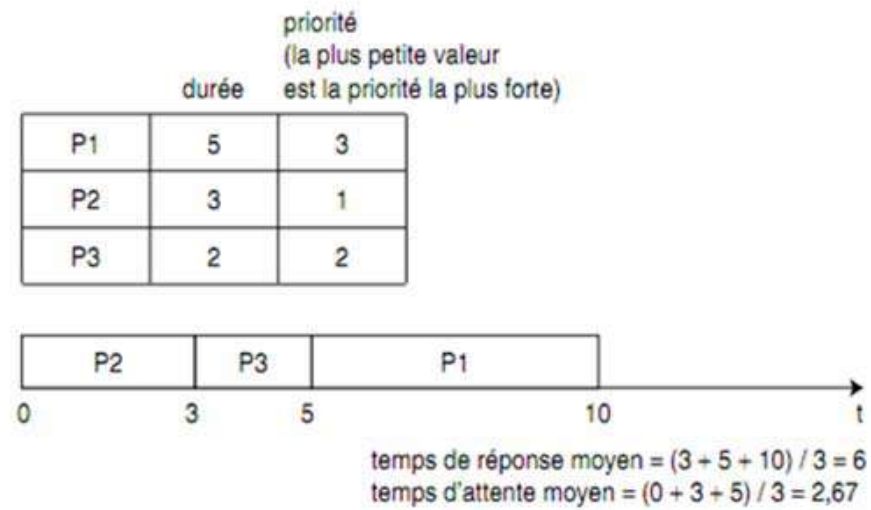
### Avantages

une gestion plus fine des processus en fonction de leur importance ou de leurs exigences de ressources

### Inconvénients

Famine (starvation) : Les processus à faible priorité peuvent attendre indéfiniment si des processus à priorité élevée continuent d'arriver

## Politique par priorité



**Exercice :**

Pour les processus du tableau, dessinez un schéma illustrant leur exécution en utilisant l'ordonnancement de priorité. Une valeur de priorité élevée correspond à une priorité plus importante.

Processus	instant d'arrivée	Cycle	Priorité
A	0,0000	4	3
B	1,0001	3	4
C	2,0001	4	6
D	3,0001	5	5

- ♦ non Préemptif
- ♦ Préemptif

## Politique du tourniquet (round robin)

- La politique mise en œuvre dans les systèmes dits en temps partagé interactifs.
- le temps est effectivement découpé en tranches nommées **quantums de temps**.
- Lorsqu'un processus est élu, il s'exécute au plus durant un quantum de temps.
  - ♦ Si le processus n'a pas terminé son exécution à l'issue du quantum de temps, il est préempté et il réintègre la file des processus prêts mais en fin de file.
- **Défaut : Choix de la valeur du quantum:**
  - ♦ Un quantum trop petit provoque **trop de basculement de processus** → abaisse l'efficacité du processeur.
  - ♦ Un quantum trop élevé **augmente le temps d'attente des processus courts interactifs**

**Un quantum entre 20 et 50 ms est souvent un compromis raisonnable**

## Politique du tourniquet (round robin)

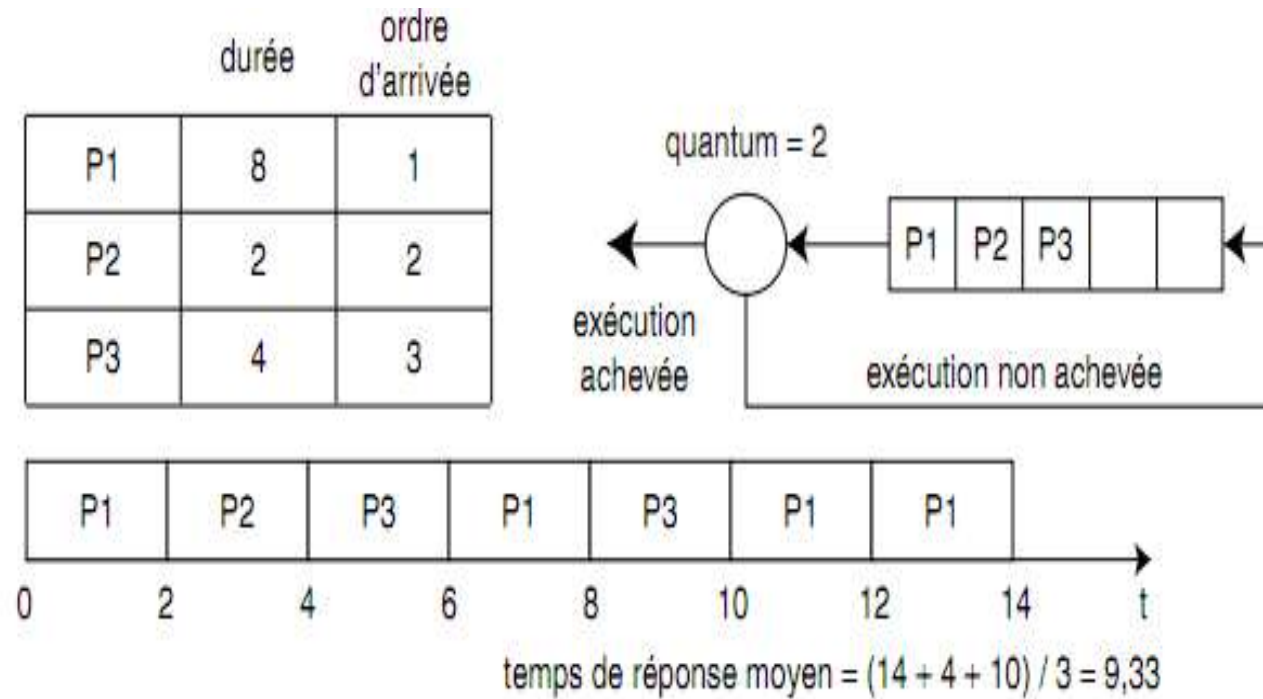


Figure 12.13 Politique du tourniquet.



## Politique du tourniquet (round robin)



### Quantum de temps

Chaque processus est exécuté pendant un quantum de temps fixe.



### Équité

Tous les processus ont la même priorité et sont servis à tour de rôle.



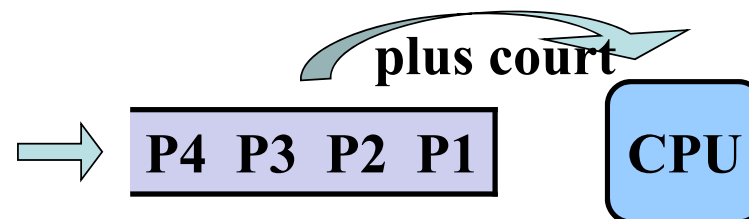
### Faible latence

Convient bien aux systèmes interactifs où la réactivité est importante.

**TEMPS RESTANT LE PLUS COURT (EN PREMIER)**  
**SRT - Shortest Remaining Time ( First )**

**avec préemption**

processus	temps d'arrivée	temps CPU
P1	0	8
P2	1	4
P3	2	9
P4	3	5



## Politiques d'ordonnancement

### **FIFO**

premier arrivé,  
premier servi

**pas de préemption**

**pénalise les processus courts**

### **SJF**

processus  
le plus court

**pas de préemption**

**pénalise les processus longs(famine)**

### **SRT**

temps restant  
le plus court

**avec préemption**

**pénalise les processus longs (famine)**

### **Priorité**

**préemptive**

**avec préemption**

**Famine encourue par le processus de faible  
priorité**

### **Round Robin**

tourniquet

**préemption (au quantum)**

**équitable ( long vs court )**

## Objectifs des Politiques d'ordonnancement

Tous les systèmes

- Équité - donner à chaque processus une part équitable du CPU
- Application de la politique - veiller à ce que la politique indiquée soit appliquée
- Équilibre - garder toutes les parties du système occupées

### **Systèmes traitement par lot**

**Le but est de maximiser le débit du processeur :  
le nombre moyen de processus traités par unité  
de temps**

### **Systèmes interactifs**

**Temps de réponse → répondre rapidement aux  
demandes  
Proportionnalité → répondre aux attentes des  
utilisateurs**

### **Systèmes temps réel**

**Le but principal est d'assurer le respect des  
contraintes de temps liées aux processus.**