

Systemes d'exploitations II

Présentation #4 : Ordonnancement des Processus

07/11/2021

Ahmed Benmoussa



Centre universitaire d'Aflou

Objectif de cette presentation

- **Rôles des ordonnanceurs**
- **Types d'ordonnanceurs**
- **L'ordonnanceur CPU.**
- **Métriques de performance.**
- **Stratégies d'ordonnancement.**
- **Algorithmes d'ordonnancement.**

Rappel: Processus

- **Etats d'un processus:** Init, prêt, en exécution, en attente, terminé
- **Opérations sur un processus**
- **fork()** est utilisé pour créer de nouveaux processus.
- **wait()** permet de retarder l'exécution d'un processus.
- **exec()** permet à un processus fils d'échapper à la similarité du processus père et d'exécuter un autre programme.

Quel processus devrait s'exécuter au suivant?

L'ordonnanceur (Scheduler)

- La partie du système d'exploitation qui choisi est appelée: **Ordonnanceur (Scheduler)**.
- L'algorithme utilisé est appelé: **Algorithme d'ordonnancement**.

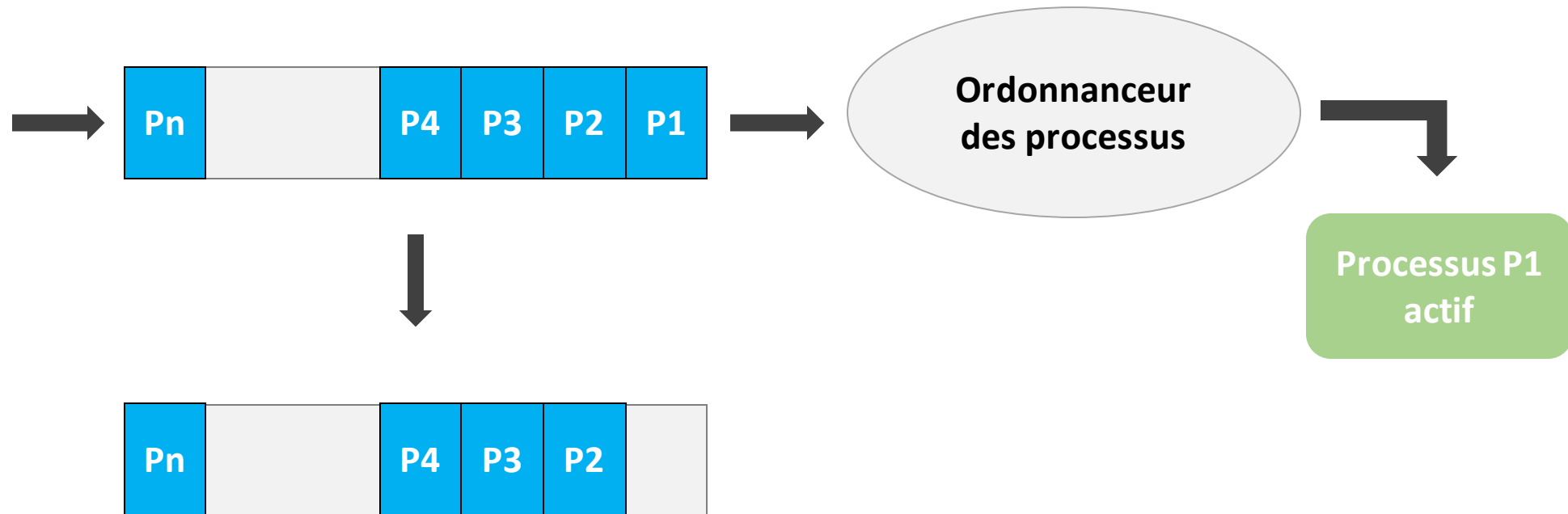
Types d'ordonnanceurs

- **CPU Scheduler:** Selectionne le prochain processus de la file d'exécution.
- **Disk Scheduler:** Selectionne la prochaine operation lecture/ecriture
- **Network Scheduler:** Selectionne le prochaine packet à être envoyer.
- **Page Replacement Scheduler** selects page to evict

Cette presentation focalise sur **l'ordonnancement des processus**
(*CPU Scheduling*)

Ordonnanceur CPU

- Son rôle consiste à choisir un processus, de la file des processus prêts, selon une stratégie d'ordonnancement et lui affecter le processeur.



Ordonnanceur CPU: Quand est ce qu'il est sollicité

- Quand un nouveau **processus est créé**.
- Quand un Processus **termine son exécution**.
- Interruption E/S
- Quand un processus **bloque** (opération E/S, semaphores, ...).

Critères de performance des algorithmes d'ordonnancement

- **Utilisation du CPU:**

Le taux d'utilisation du CPU doit être le plus élevé possible.

- **Temps de résidence ou temps de séjour (Turnaround time)**

C'est le temps passé par le processus dans le système.

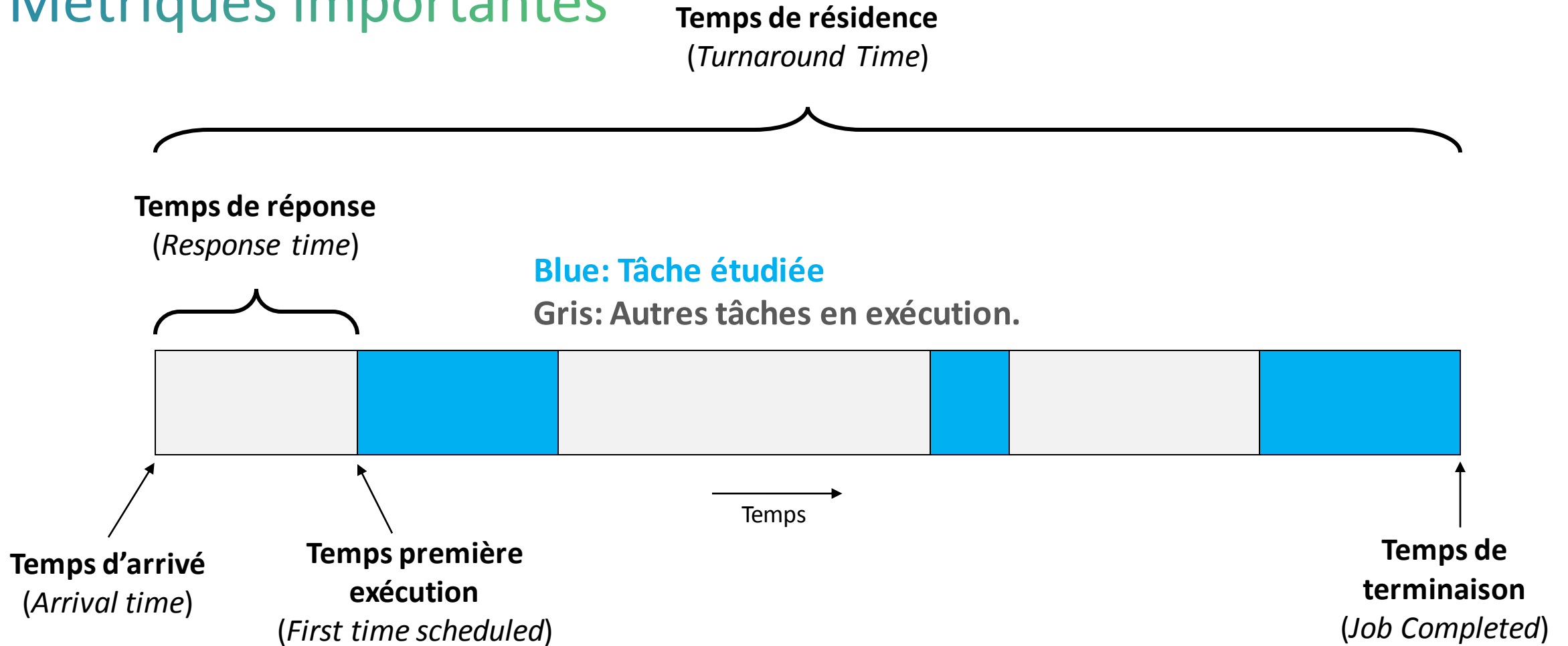
- **Temps d'attente (Waiting time): Attente du processeur**

Le temps passé en attente du CPU.

- **Temps de réponse (Response time)**

Le temps qui sépare le moment où le processus envoie une requête et le début de la réponse à cette requête.

Métriques importantes



- **Temps d'exécution**: somme des périodes **bleues**.
- **Temps d'attente**: somme des périodes **grises**.
- **Temps de residence**: somme des deux.

L'ordonnanceur parfait (Perfect Scheduler)

- Minimiser **le temps de réponse** de chaque job.
- Minimiser **le temps de residence** de chaque job.
- Maximiser le **debit global**.
- Maximiser **l'utilisation des ressources**: faire occuper toutes les ressourcees.
- Respecter tous **les délais**.
- **Juste**: tous les jobs progressent.
- **Envy-Free**: aucun job veut avoir la place d'un autre.
- **Zero overhead**.

Stratégies d'ordonnancement

Deux catégories:

1. Non Préemptive

- Le processus choisi n'est pas interrompu au cours de son exécution sauf:
- s'il est bloqué (E/S, attente d'un autre processus).
- Quand il relâche volontairement le CPU.

2. Préemptive

- Le processus choisi s'exécute pendant une durée prédéfinie.
- Le processus en exécution est suspendu et l'ordonnanceur choisi un autre processus (s'il existe).
- Nécessite interruption d'horloge.

Algorithmes d'ordonnement

- First Come First Served (FCFS) ou First In First Out (FIFO)
- Shortest Job First (SJF)
- Earliest Deadline First (EDF)
- Shortest Remaining Time Next (SRTN)
- Round Robin (RR)
- Algorithmes avec priorité
- Multi-Level Feedback Queue (MLFQ)

Algorithmes d'ordonnement

- **First Come First Served (FCFS) ou First In First Out (FIFO)**
- Shortest Job First (SJF)
- Earliest Deadline First (EDF)
- Shortest Remaining Time Next (SRTN)
- Round Robin (RR)
- Algorithmes avec priorité
- Multi-Level Feedback Queue (MLFQ)

FCFS (Premier arrivé premier servis)

- C'est un algorithme d'ordonnancement non préemptive
- Consiste à allouer le processeur au premier processus de la file des processus prêts.
- Le processeur ne peut être retiré au processus que s'il le libère volontairement.

FCFS: 1^{er} exemple

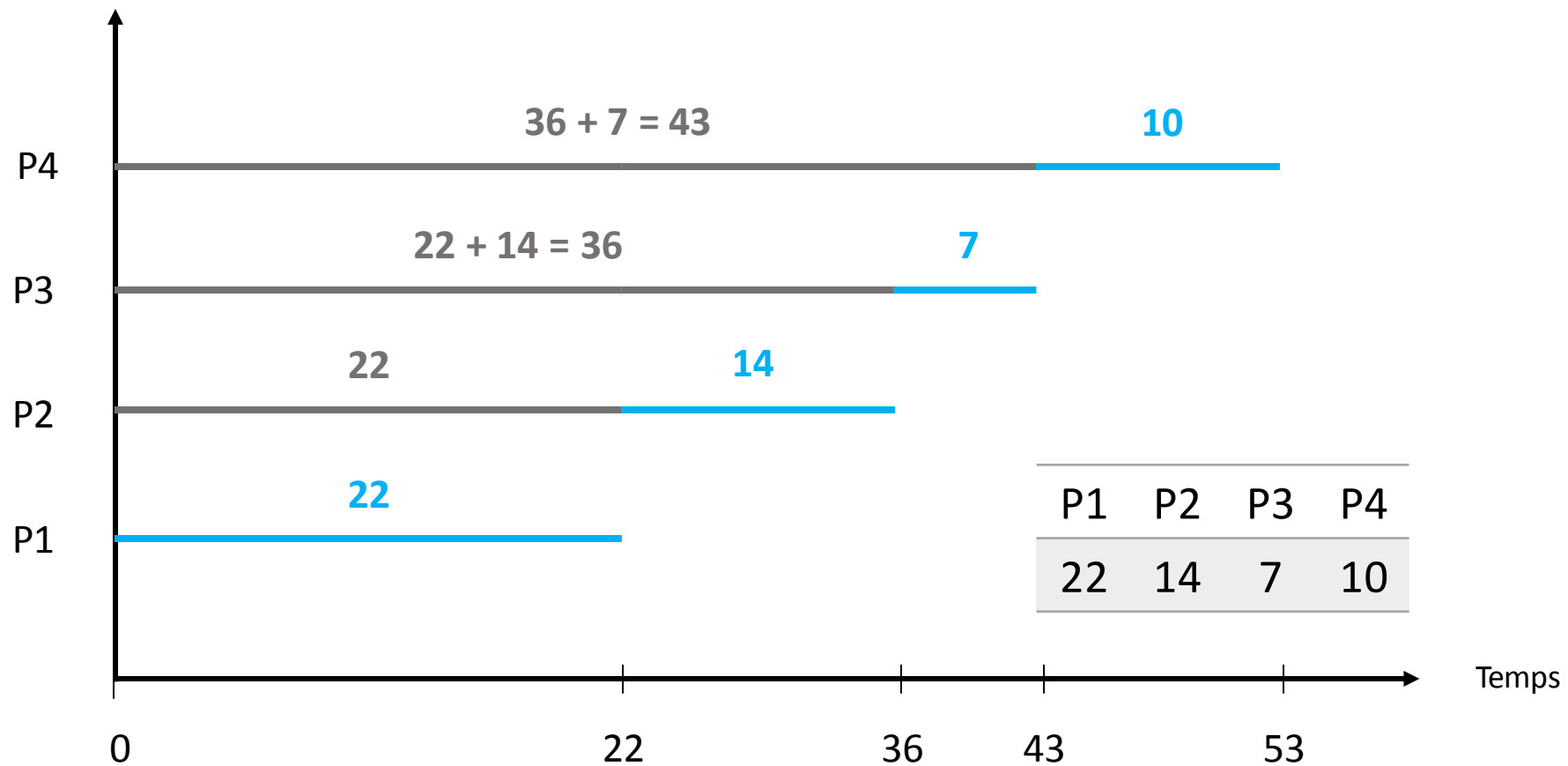
- **Exemple** : à l'instant **t** la file des processus à l'état prêt contient 4 processus : P0, P1, P2 et P3



Processus	P1	P2	P3	P4
Temps d'exécution estimé	22	14	7	10

FCFS: 1^{er} exemple

- Représentation de l'exécution des processus à l'aide d'un **diagramme de Gantt**



FCFS: 1^{er} exemple (Résultat)

Processus	Temps estimé	Temps de résidence	Temps d'attente
P0	22	22	$22 - 22 = 0$
P1	14	$22 + 14 = 36$	$36 - 14 = 22$
P2	7	$36 + 7 = 43$	$43 - 7 = 36$
P3	10	$43 + 10 = 53$	$53 - 10 = 43$
Moyenne		38,5	25,25

FCFS: 2^{ème} exemple

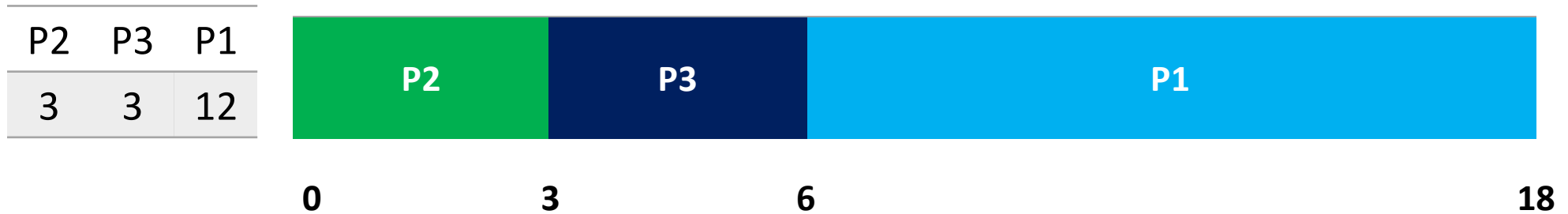
- Les processus P1, P2 et P3 arrivent au **même moment**.
- Dans ce cas la, il peuvent être ordonné dans n'importe quel ordre.

Processus	P1	P2	P3
Temps d'exécution estimé	12	3	3

FCFS: 2^{ème} exemple



$$\text{Temps de résidence} = (12 + 15 + 18)/3 = 15$$



$$\text{Temps de résidence} = (3 + 6 + 18)/3 = 9$$

FCFS: Roundup

- Simple.
 - Coût réduits.
 - Pas de famine (starvation).
-
- Le temps de résidence moyen varie selon l'ordonnancement.
 - Pas adapté au Jobs interactifs.

Comment minimiser le temps de résidence moyen?

Algorithmes d'ordonnement

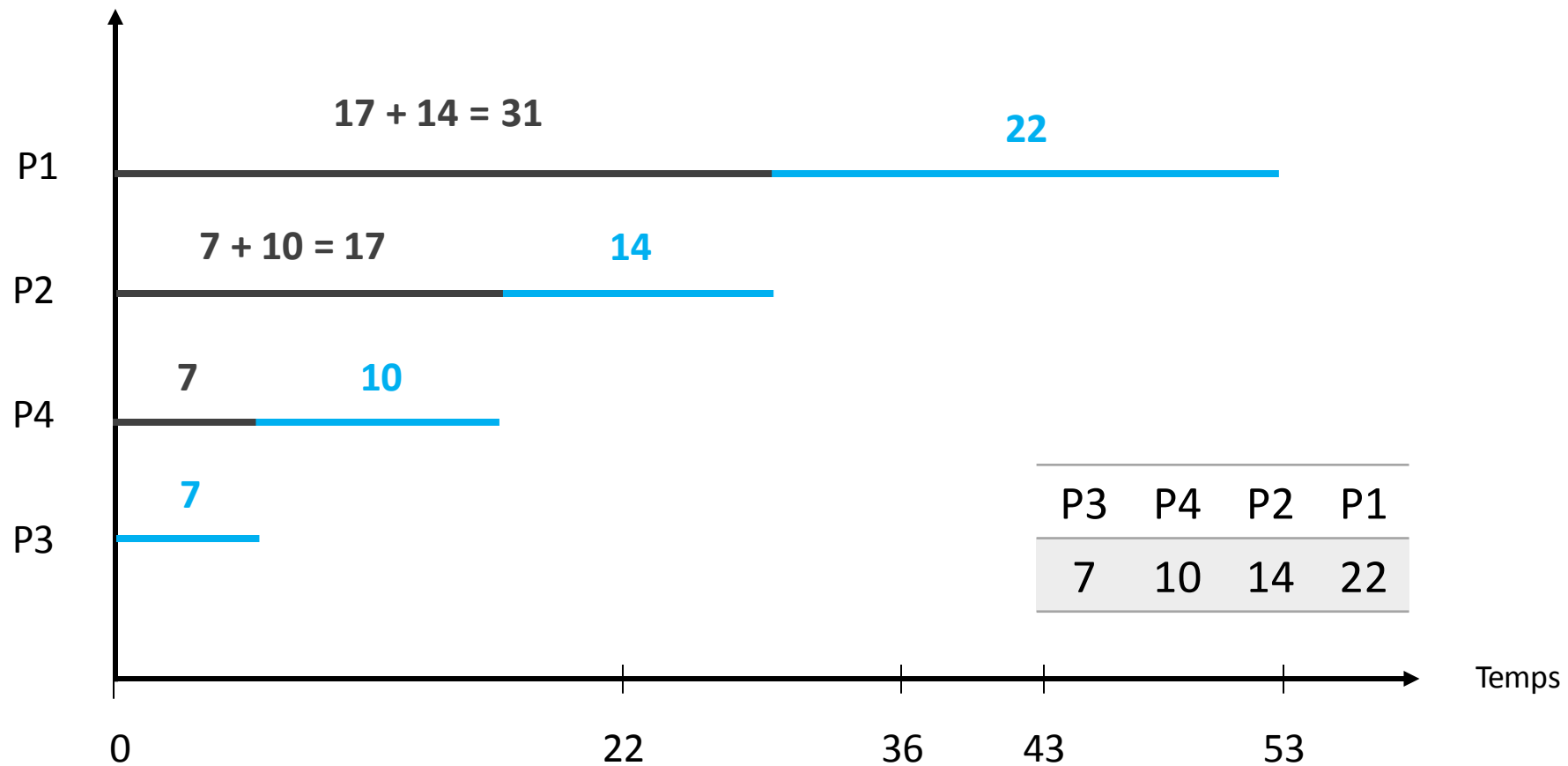
- First Come First Served (FCFS) ou First In First Out (FIFO)
- **Shortest Job First (SJF)**
- Earliest Deadline First (EDF)
- Shortest Remaining Time Next (SRTN)
- Round Robin (RR)
- Algorithmes avec priorité
- Multi-Level Feedback Queue (MLFQ)

SJF: Le plus court d'abord

- L'algorithme consiste à allouer le Job **le plus court en premier**.
- La priorité d'un processus est inversement proportionnelle au temps CPU demandé.
- SJF est un cas particulier des algorithmes à priorités:
 - $\text{Priorité} = 1/\text{temps_CPU}$

SJF: exemple

- Représentation de l'exécution des processus à l'aide d'un **diagramme de Gantt**



SJF: exemple (Résultat)

Processus	Temps estimé	Temps de résidence	Temps d'attente
P3	7	7	$7 - 7 = 0$
P4	10	$7 + 10 = 17$	$17 - 10 = 7$
P2	14	$17 + 14 = 31$	$31 - 14 = 17$
P1	22	$31 + 22 = 53$	$53 - 22 = 31$
Moyenne		27	13,75

FCFS :

Temps de résidence = **38,5**

Temps d'attente = **25,25**

SJF: Roundup

- Temps de résidence moyen optimal.
- Le temps d'exécution doit être connu.
- Peut affamer les long Jobs.

Algorithmes d'ordonnement

- First Come First Served (FCFS) ou First In First Out (FIFO)
- Shortest Job First (SJF)
- **Earliest Deadline First (EDF)**
- Shortest Remaining Time Next (SRTN)
- Round Robin (RR)
- Algorithmes avec priorité
- Multi-Level Feedback Queue (MLFQ)

EDF: Date limite la plus proche

- L'algorithme consiste à trier les Jobs selon leur date (temps) limite.
- Cet algorithme n'a pas besoin de connaître le temps d'exécution des Jobs.

EDF: Roundup

- Idéal pour les Jobs avec data limite.
- Pas de famine.
- N'utilise pas d'autres métriques
- Ne peut pas décider quand exécuter un Job sans date limite.

Algorithmes d'ordonnement

- First Come First Served (FCFS) ou First In First Out (FIFO)
- Shortest Job First (SJF)
- Earliest Deadline First (EDF)
- **Shortest Remaining Time Next (SRTN)**
- Round Robin (RR)
- Algorithmes avec priorité
- Multi-Level Feedback Queue (MLFQ)

SRTN: Le temps restant le plus cours

- Algorithme préemptif du *Shortest Job First*
- L'algorithme choisi le Processus ayant le temps d'exécution restant le plus cours.
- Si un nouveau Job a besoin de moins de temps qu'un processus en exécution, ce dernier sera suspendu et le nouveau Job démarré.

SRTN: Roundup

- Idéal pour les petits Jobs.
- Nécessite de connaître le temps d'exécution.

Algorithmes d'ordonnement

- First Come First Served (FCFS) ou First In First Out (FIFO)
- Shortest Job First (SJF)
- Earliest Deadline First (EDF)
- Shortest Remaining Time Next (SRTN)
- **Round Robin (RR)**
- Algorithmes avec priorité
- Multi-Level Feedback Queue (MLFQ)

Round Robin: Tourniquet

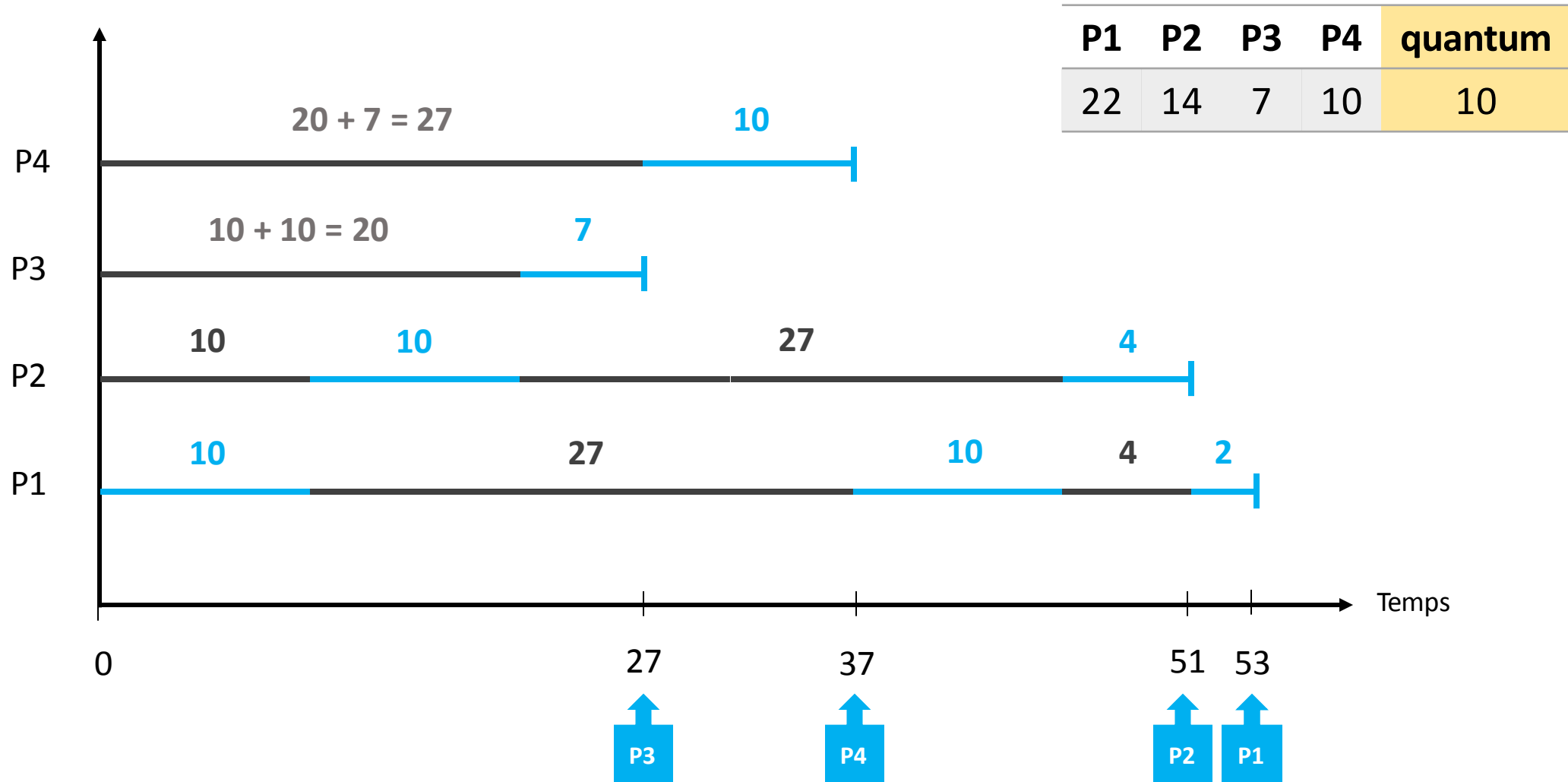
- Chaque processus s'exécute pendant un temps (**quantum**).
- quantum de temps = Une période de temps pré-configuré.
- Le processeur est alloué au premier processus de la file des 'processus prêts'
- Si le processus n'a pas terminé son exécution, il est recyclé dans la file des 'processus prêts'.
- Le processeur est alloué à un autre processus :
 - interruption horloge (fin du quantum),
 - attente de ressource physique ou logique (E/S),
 - Fin d'exécution du processus (fin normale ou erreur).

Round Robin: Tourniquet

Quel est le quantum idéal?

- Quantum très grand: équivaut à l'algorithme FIFO.
- Quantum très court: perte de temps (interruptions).
- Typical quantum: 100X le coût de commutation des contextes.

RR: Exemple



RR: Exemple (résultats)

Processus	Temps estimé	Temps de résidence	Temps d'attente
P1	22	53	$53 - 22 = 31$
P2	14	51	$51 - 14 = 37$
P3	7	27	$27 - 7 = 20$
P4	10	37	$37 - 10 = 27$
Moyenne		42	28,75

Algorithmes d'ordonnement

- First Come First Served (FCFS) ou First In First Out (FIFO)
- Shortest Job First (SJF)
- Earliest Deadline First (EDF)
- Shortest Remaining Time Next (SRTN)
- Round Robin (RR)
- **Algorithmes avec priorité**
- Multi-Level Feedback Queue (MLFQ)

Algorithmes d'ordonnement avec priorité

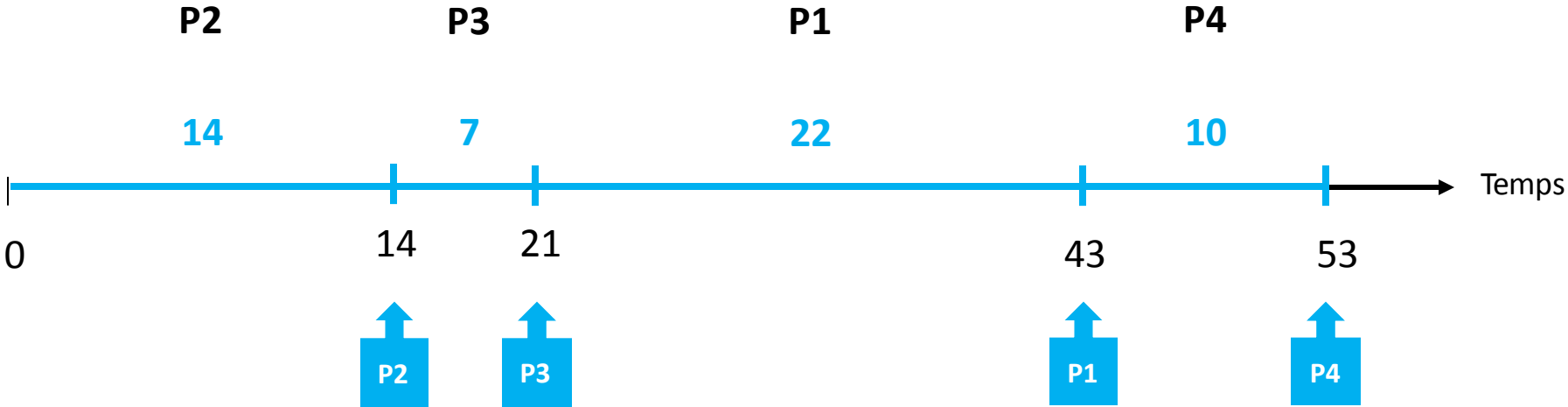
- Assigné une **valeur numérique** a chaque processus et les classer de manière ascendante.
- La priorité est assigné au processus selon leur importance dans le système.
- Deux type: **Priorité statique** et **priorité dynamique**.

Algorithmes d'ordonnement avec priorité

- **Priorité statique:** La valeur attribuée au processus à sa création ne changera pas tout au long de son exécution.
- **Priorité dynamique:** La valeur attribuée initialement au processus peut évoluer, par exemple:
 - En fonction du temps d'exécution;
 - En fonction du nombre d'E/S;
 - En fonction du nombre de ressources consommées;
 - ...
- L'évolution de la priorité peut être dans les deux sens : **positif** ou **négatif**

Algorithmes d'ordonnancement avec priorité: Exemple

Processus	P1	P2	P3	P4
Temps d'exécution	22	14	7	10
priorité	3	1	2	3



Algorithmes d'ordonnancement avec priorité: Exemple

Processus	Temps estimé	Temps de résidence	Temps d'attente
P1	22	43	$43 - 22 = 21$
P2	14	14	$14 - 14 = 0$
P3	7	21	$21 - 7 = 14$
P4	10	53	$53 - 10 = 43$
Moyenne		32,75	19,5

Algorithmes d'ordonnement

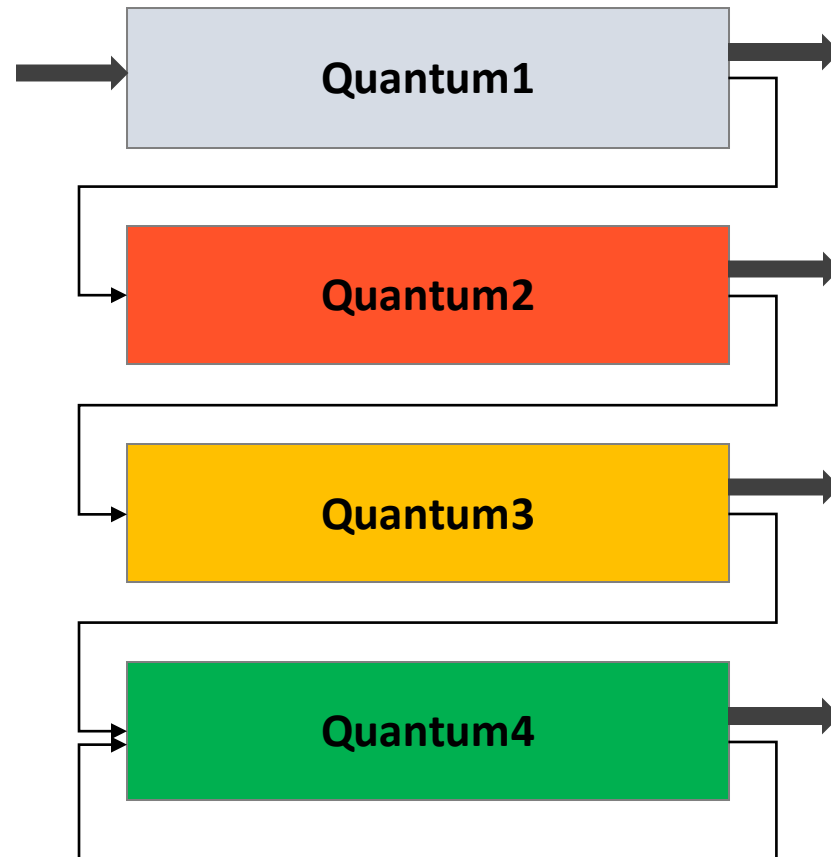
- First Come First Served (FCFS) ou First In First Out (FIFO)
- Shortest Job First (SJF)
- Earliest Deadline First (EDF)
- Shortest Remaining Time Next (SRTN)
- Round Robin (RR)
- Algorithmes avec priorité
- **Multi-Level Feedback Queue (MLFQ)**

MLFQ: Ordonnancement a files multi-niveau

- Multiple levels of RR queue
- Multiples niveau de files (tourniquet)
- Les Job commence par la file la plus en haut .
- A chaque utilisation du quantum, le Job est déplacé vers le bas.
- Un processus de la file f_i n'est servi que si toutes les files de rang inférieur à i sont vides.

MLFQ: Ordonnancement a files multi-niveau

**Plus haute
priorié**

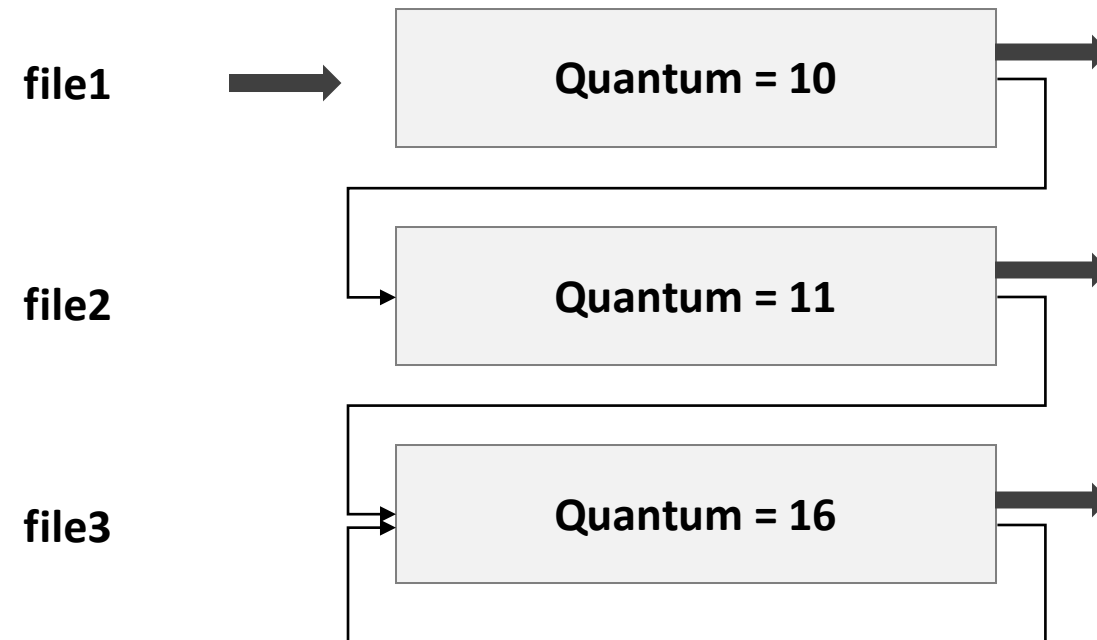


**Plus faible
priorié**

$\text{quantum1} < \text{quantum2} < \text{quantum3} < \text{quantum4}$

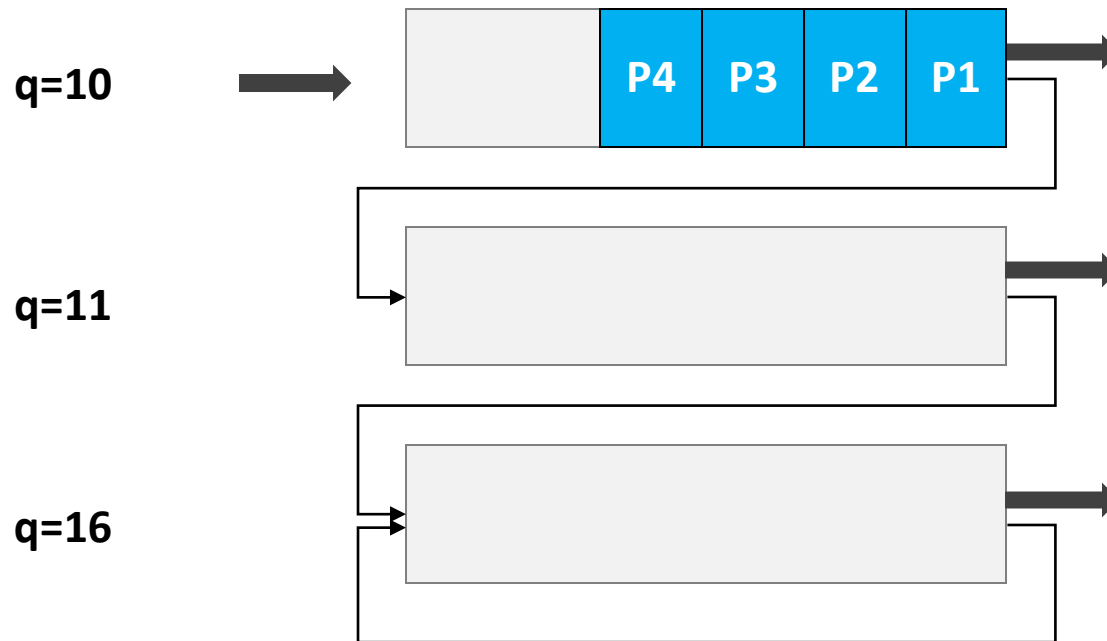
MLFQ: Exemple

Processus	P1	P2	P3	P4
Temps d'exécution	22	14	7	10



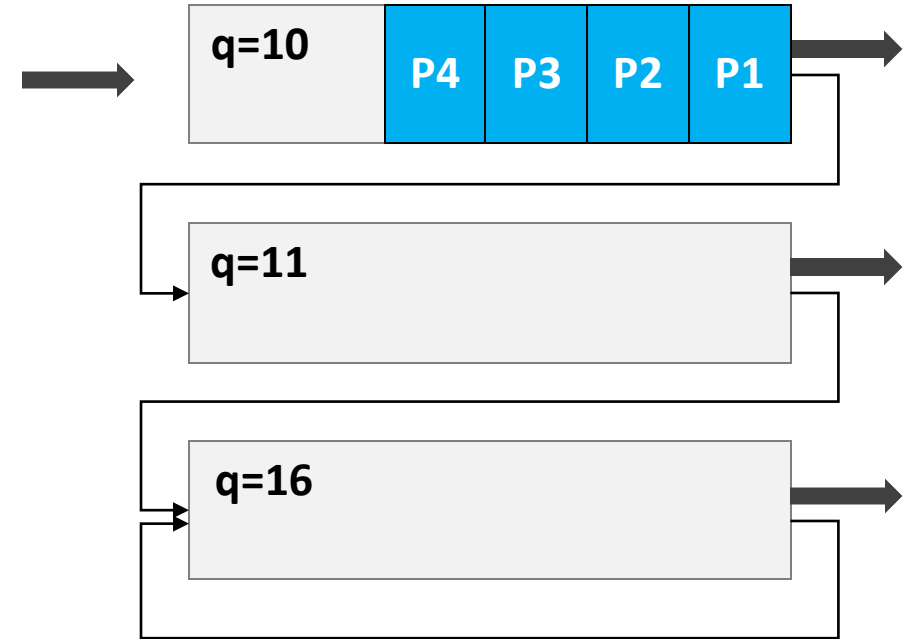
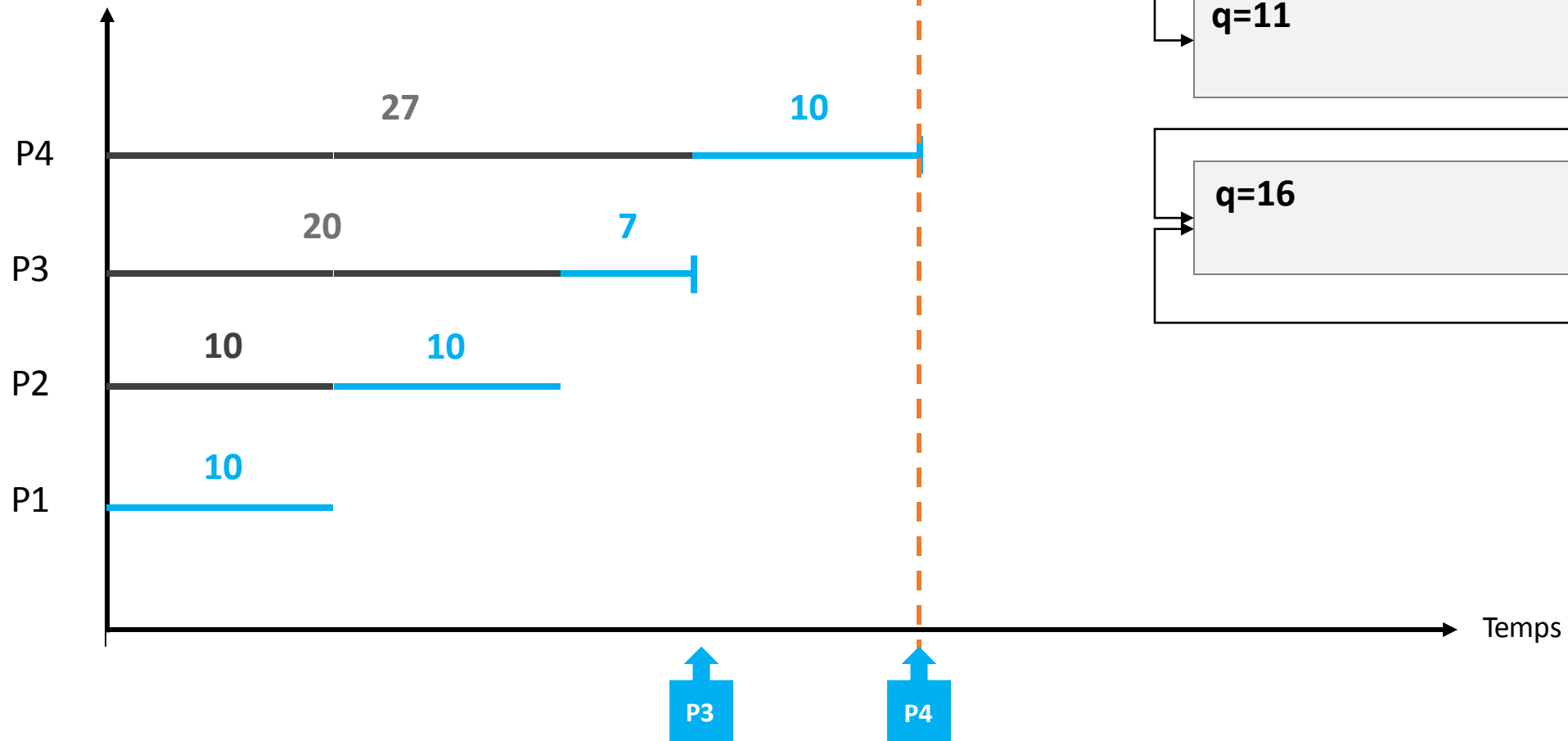
MLFQ: Exemple

Processus	P1	P2	P3	P4
Temps d'exécution	22	14	7	10

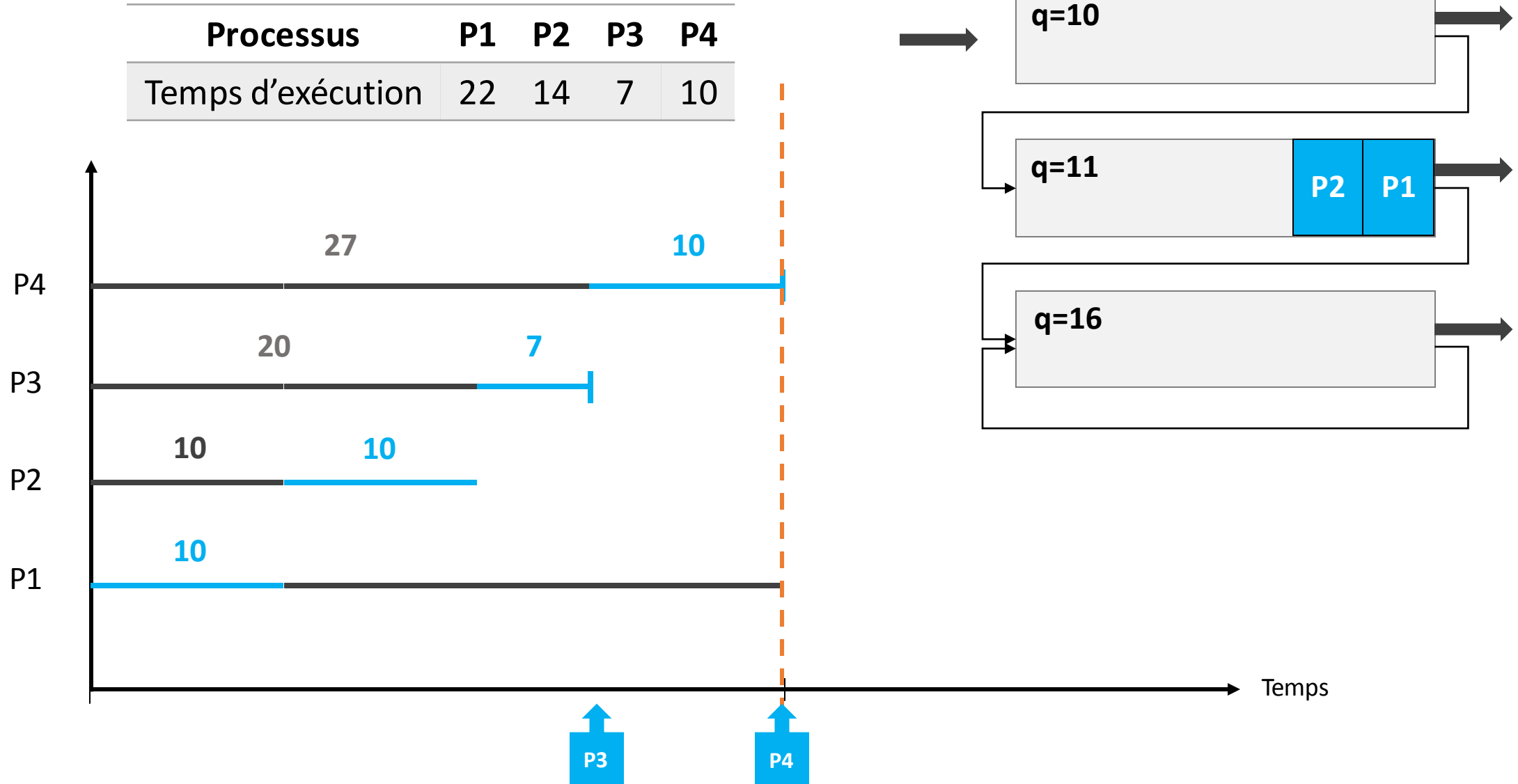


MLFQ: Exemple

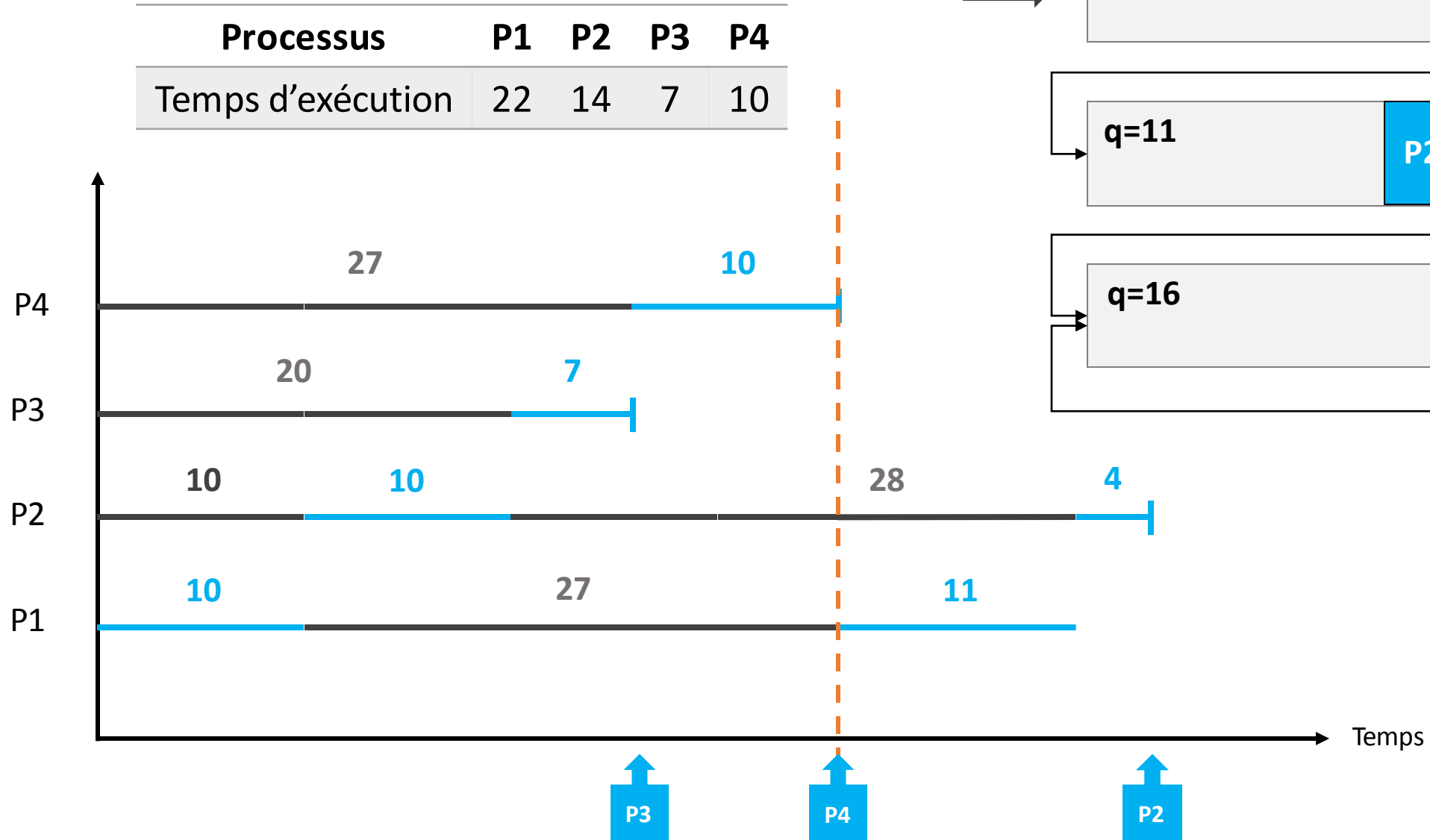
Processus	P1	P2	P3	P4
Temps d'exécution	22	14	7	10



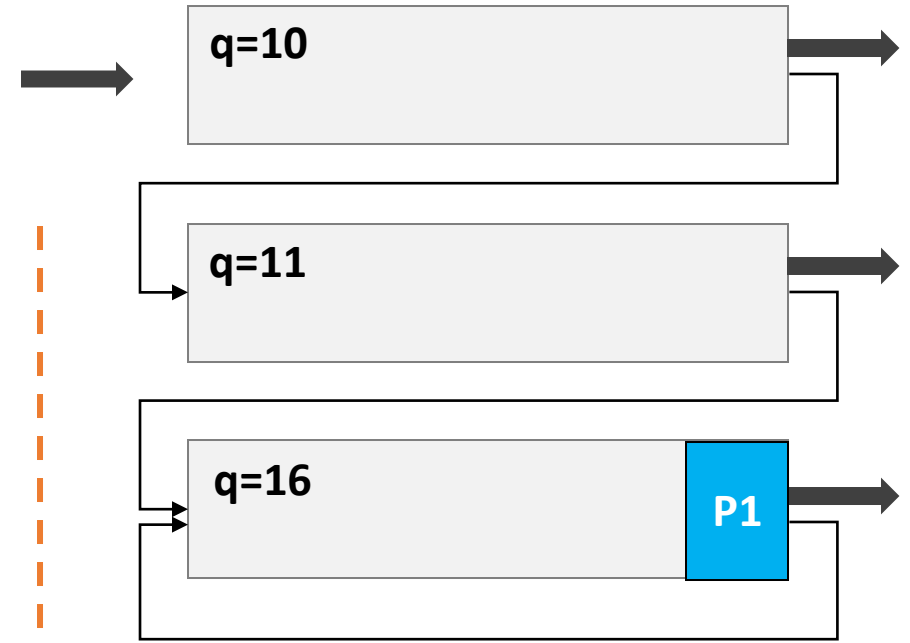
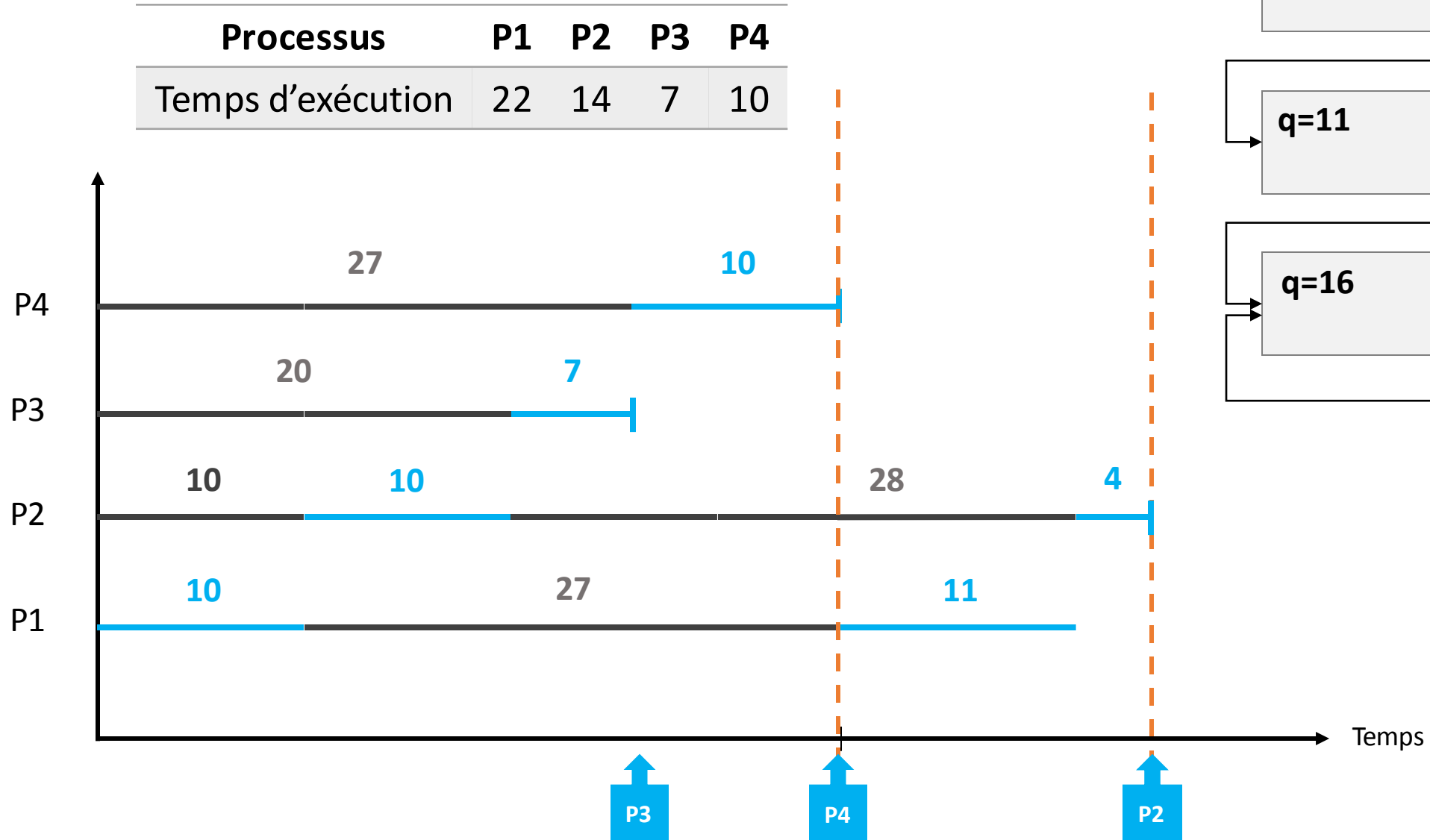
MLFQ: Exemple



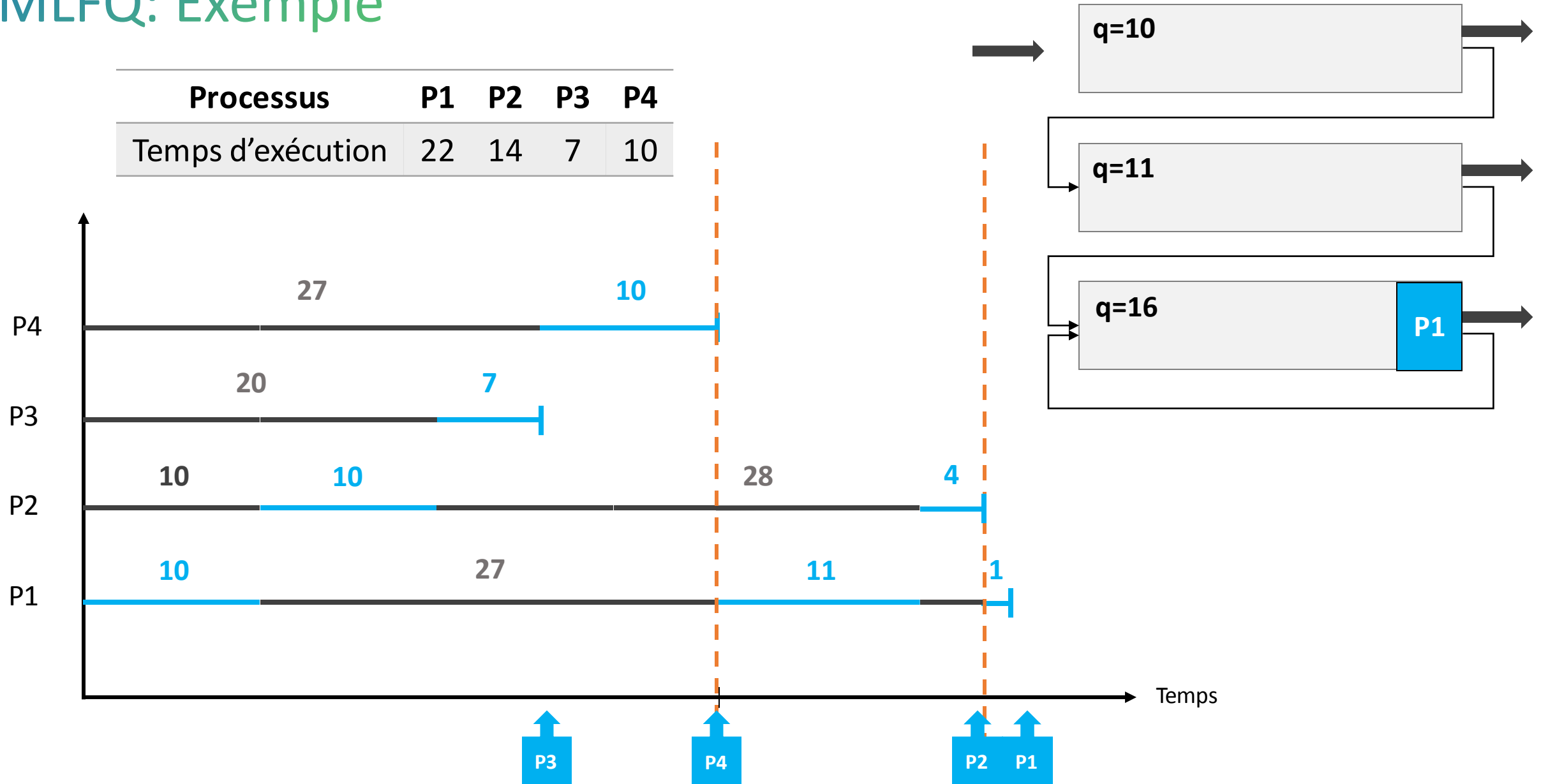
MLFQ: Exemple



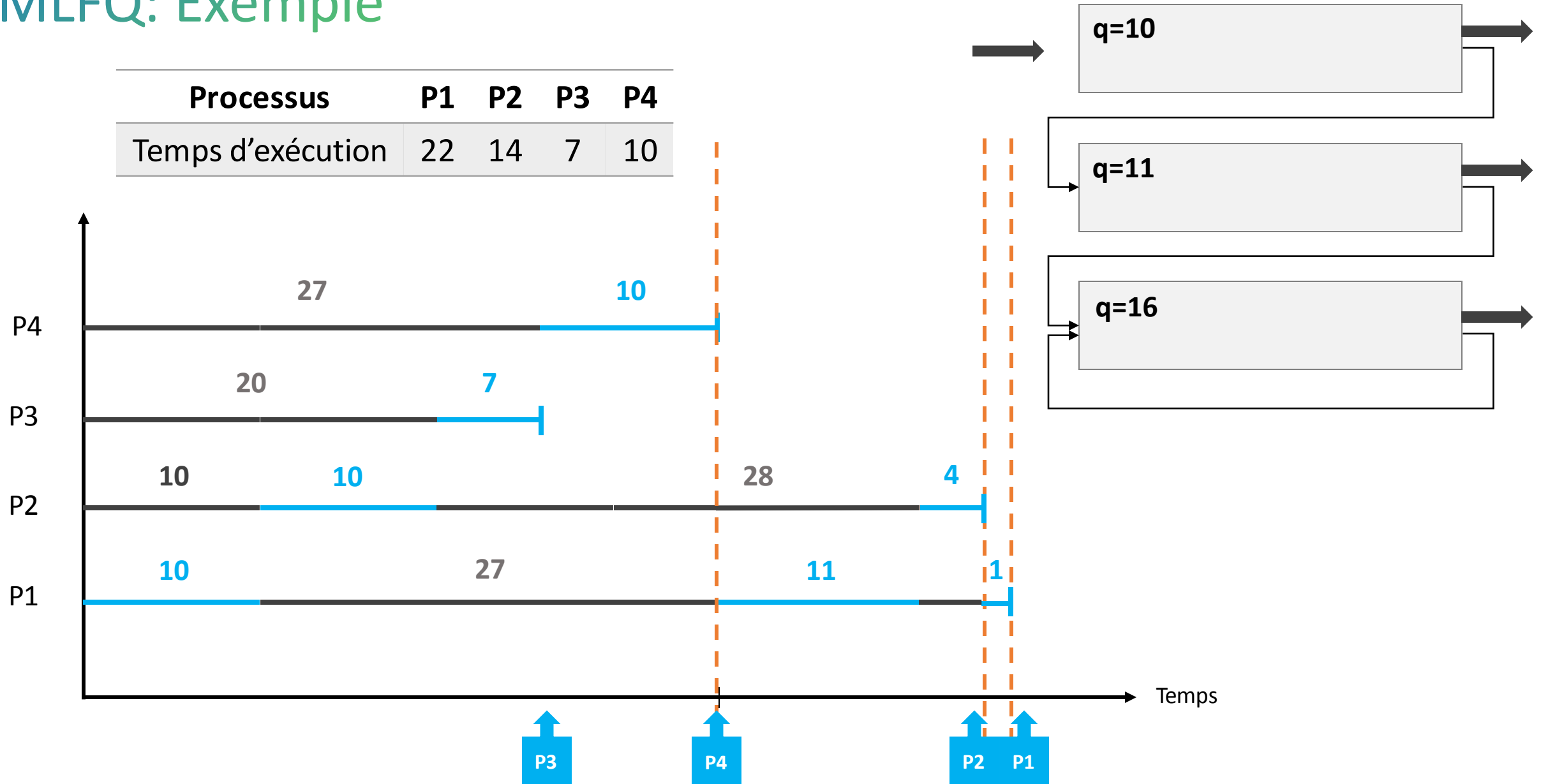
MLFQ: Exemple



MLFQ: Exemple



MLFQ: Exemple



Algorithmes d'ordonnancement avec priorité: Exemple

Processus	Temps estimé	Temps de résidence	Temps d'attente
P1	22	51	$51 - 22 = 29$
P2	14	50	$50 - 14 = 36$
P3	7	27	$27 - 7 = 20$
P4	10	37	$37 - 10 = 27$
Moyenne		41,25	28

Conclusion

- **L'ordonnanceur CPU.**
- **Métriques de performance:** temps de résidence, temps d'attente, temps de réponse, ...
- **Stratégie d'ordonnancement:** préemptif et non-préemptifs
- **Algorithme d'ordonnancement:** FIFO, SJF, RR, MLFQ, ...