# Systèmes d'Exploitation - ENSIN6U3

## Ordonnancement

Leonardo Brenner [1] Jean-Luc Massat [2]

[1] Leonardo.Brenner@univ-amu.fr

[2] Jean-Luc.Massat@univ-amu.fr

Aix-Marseille Université

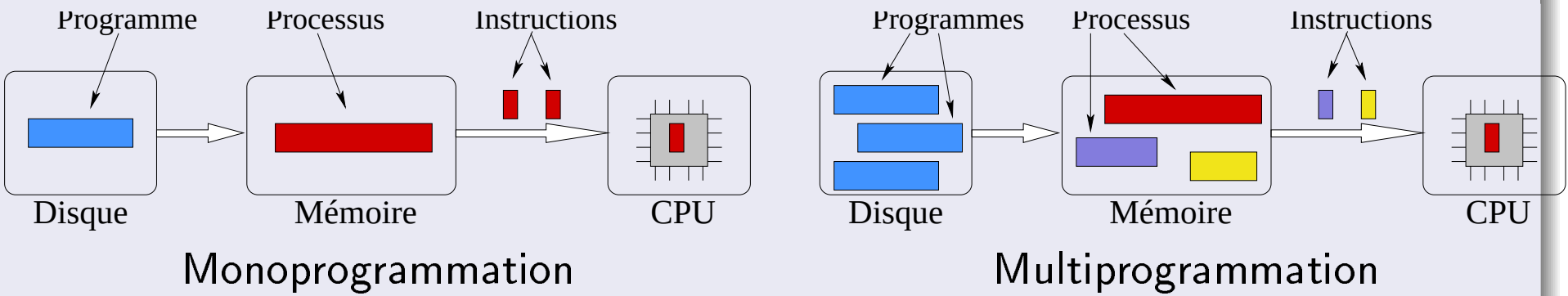
Faculté des Sciences

# Table de matière

# Table de matière

# Problématique

## Exécution de processus




## Cas d'un système multiprogrammé

- Plusieurs processus en mémoire, un seul à la fois accède au CPU ;
- Comment choisir le processus qui a accès au CPU ? Comment changer cet accès au cours de l'exécution ?

# Ordonnancement de processus

## *Définition* : Ordonnanceur

L'ordonnanceur est un algorithme qui va élire le processus qui a accès à la CPU. Ce processus a le « privilège »d'accéder à la CPU pendant un intervalle de temps.

## Perte de la CPU

Il y a perte de la CPU dans trois cas :

- interruption extérieur,
- déroutements,
- appels système.

# Objectifs et critères

## Objectifs d'un bon ordonnancement

- équitable,
- débit maximum,
- rationaliser la gestion des ressources,
- favoriser les « bons »processus,
- améliorer les temps de réponse moyen.

## Critères d'ordonnancement

- le taux d'E/S,
- le taux d'utilisation de la CPU,
- le type et la priorité du processus,
- temps CPU cumulé,
- temps CPU restant,
- taux de réquisition.

## Système interactif

On cherche à optimiser le temps de réponse court pour répondre à l'utilisateur

# Commutation de contexte

## *Définition* : Commutation de contexte

Une commutation de contexte consiste à sauvegarder l'état d'un processus et à restaurer l'état d'un autre processus.

## Ordonnanceur non préemptif

On sélectionne un processus qui s'exécute jusqu'à ce qu'il libère volontairement le processeur.

## Ordonnanceur préemptif (avec réquisition)

- On sélectionne un processus qui s'exécute pendant un intervalle de temps ;
- Si le processus est toujours en cours d'exécution après cet intervalle de temps, il est suspendu et un autre processus est choisi

# Type d'allocation

## Quel type d'allocation

- *long terme* : variation du degré de multi-programmation par *swap-ping*,
- *moyen terme* : détermine et classe les processus prêts,
- *court terme* : répartition de la CPU.

# Mesures de performances (1/2)

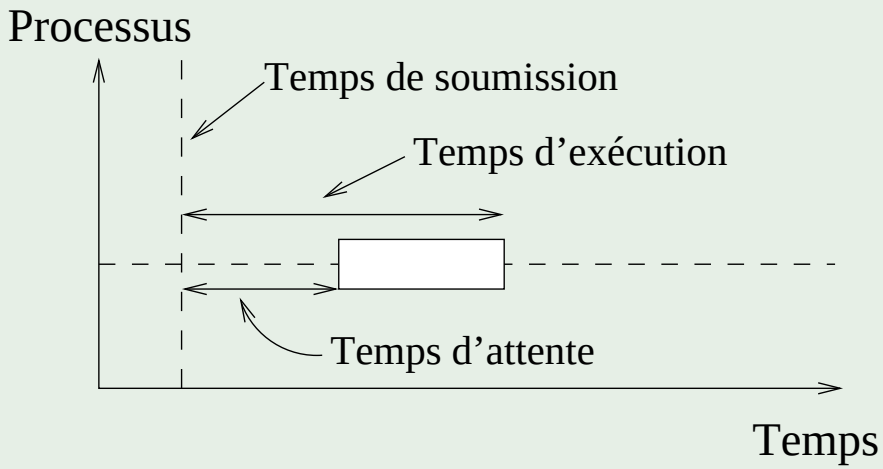
**Définition : Temps d'attente ou réponse (d'un processus)**

Temps nécessaire pour qu'un processus soit élu pour la première fois par l'ordonnanceur depuis sa soumission.

**Définition : Temps d'exécution (d'un processus)**

Temps écoulé entre la soumission et la fin de l'exécution du processus.

**Illustration**



# Mesures de performances (2/2)

## Performances d'un algorithme d'ordonnancement

- Temps moyen d'attente : moyenne de tous les temps d'attente de chaque processus ;
- Temps moyen d'exécution : moyenne de tous les temps d'exécution de chaque processus

## Quelques algorithmes simples

- First In First Out (FIFO)
- Shortest Job First (SJF)
- Shortest Remaining Time (SRT)
- Highest Response Ratio Next (HRN)
- Round Robin (tourniquet) (RR)
- Files de priorité
- Files de priorité dynamique

# Table de matière

# First In First Out

## Description de l'algorithme *First In First Out*

Les processus sont exécutés dans l'ordre de soumission.

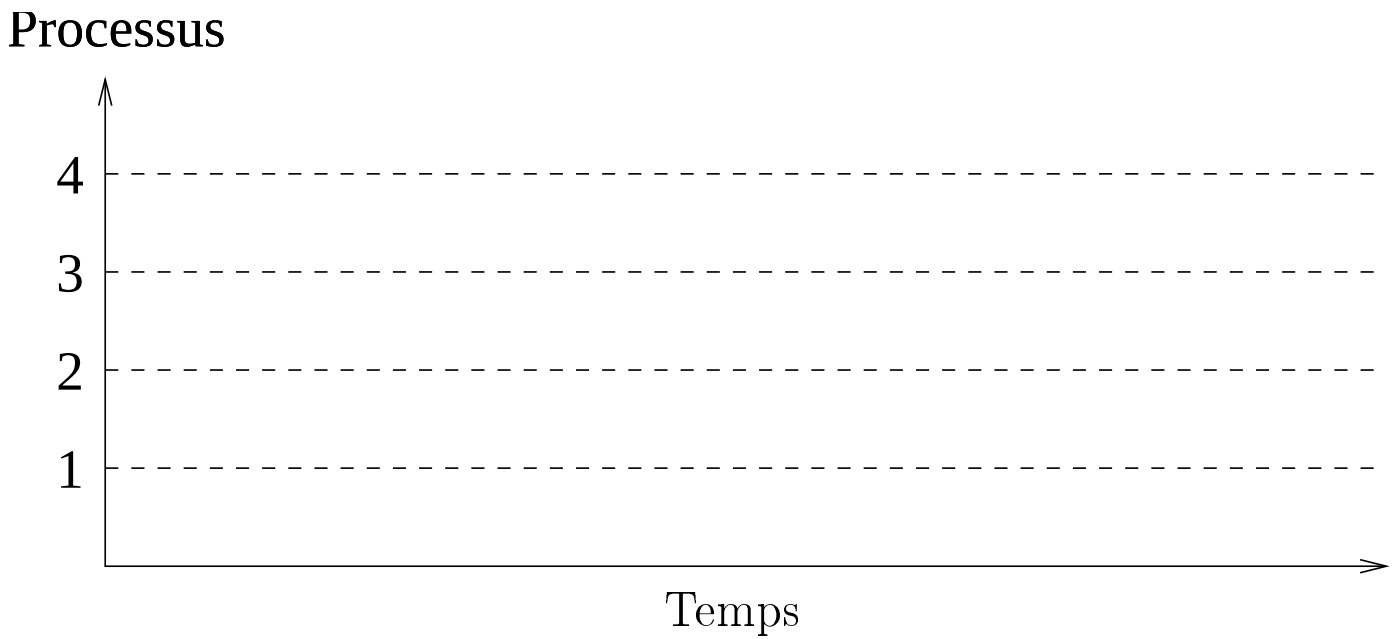
## Propriétés de l'algorithme

- Pas de notion de priorité sur les processus ;
- Non préemptif (pas de réquisition) ;
- Pas de connaissance sur la durée des processus ;
- Croissance rapide du temps d'attente.

# First In First Out : exemple

## Données des processus

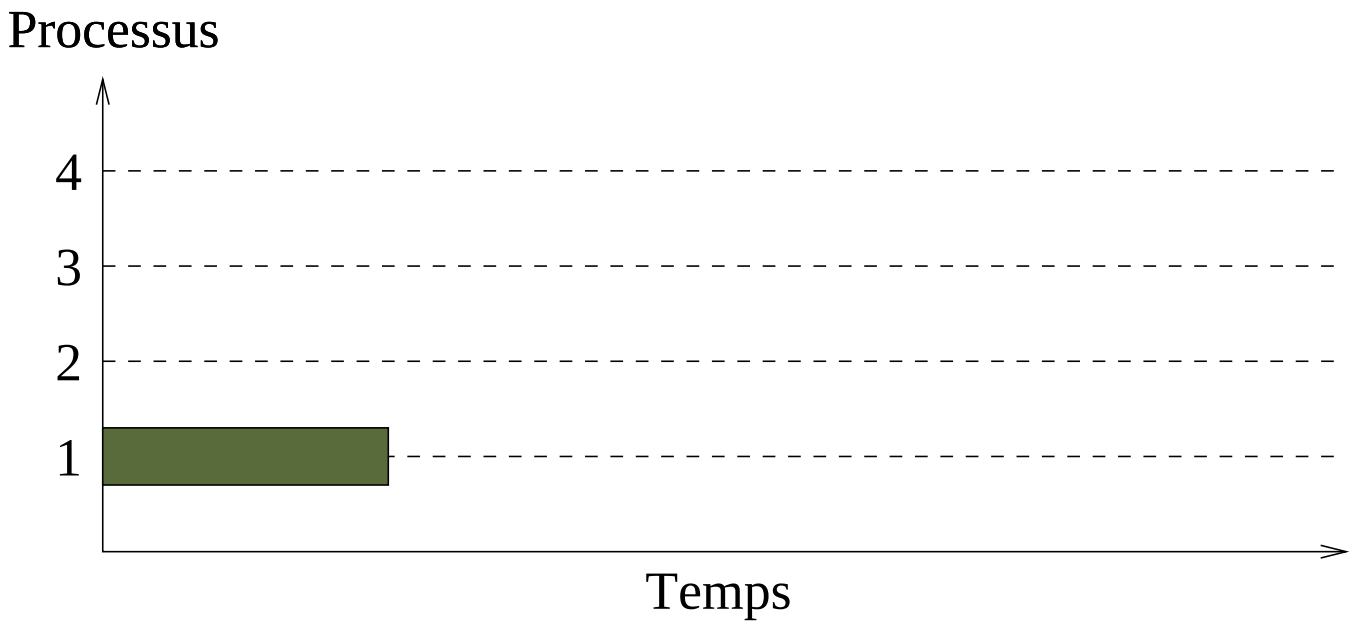
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# First In First Out : exemple

## Données des processus

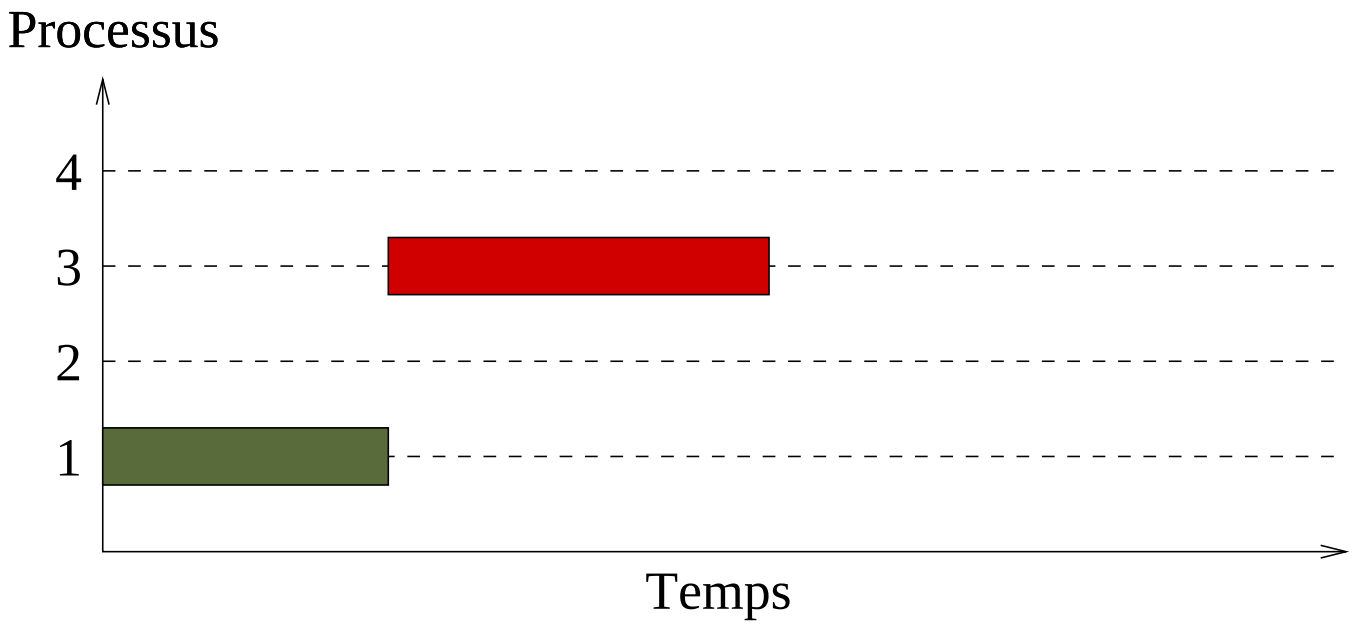
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# First In First Out : exemple

## Données des processus

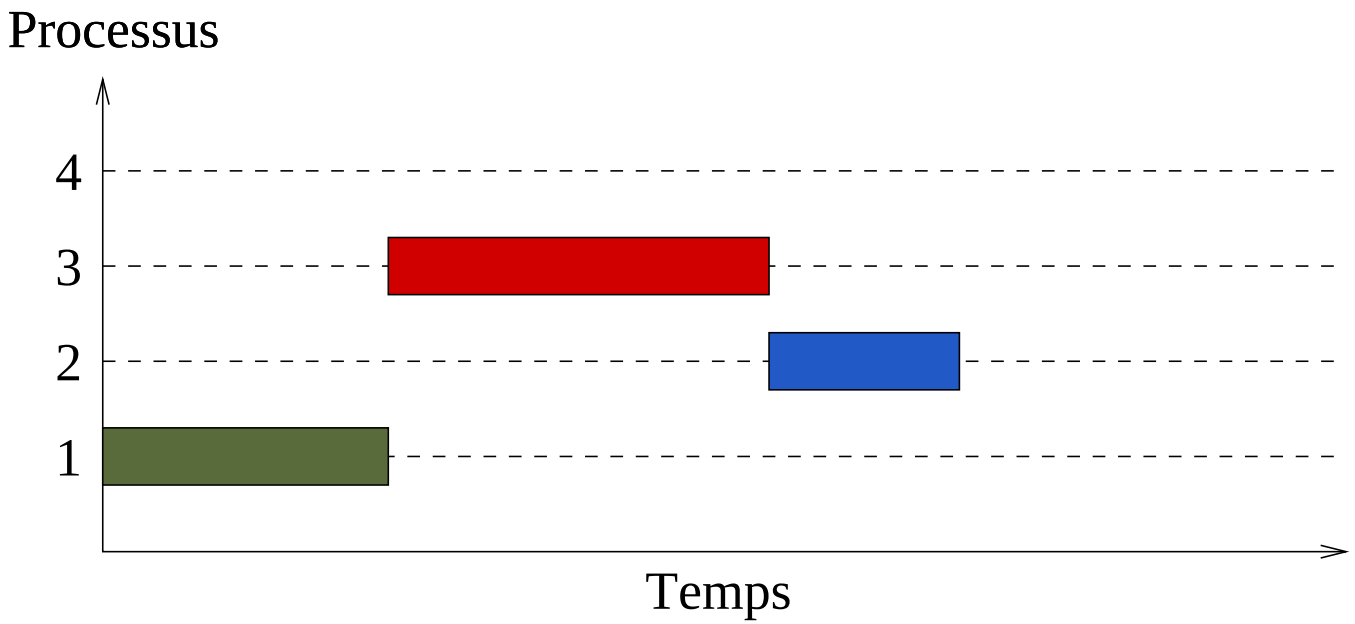
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# First In First Out : exemple

## Données des processus

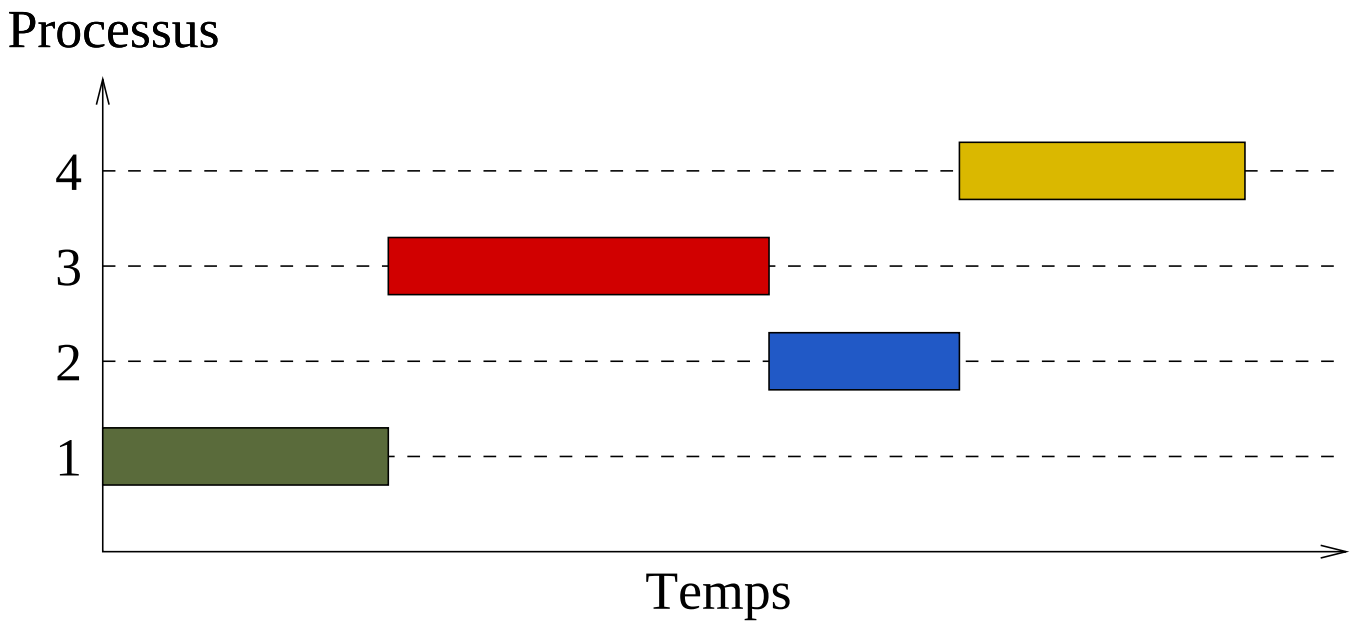
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# First In First Out : exemple

## Données des processus

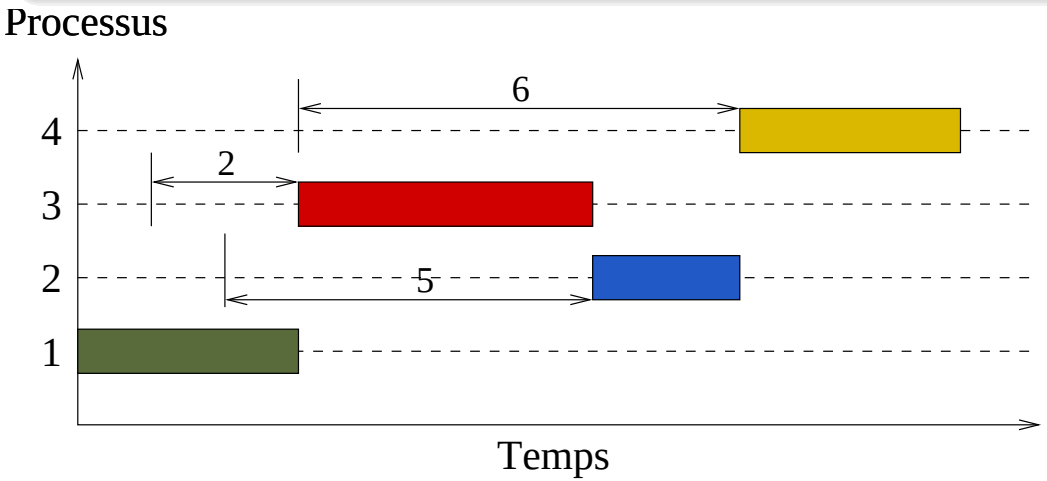
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# First In First Out : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |




| Processus | Temps d'attente | Temps d'exécution |
|---|---|---|
| 1 | 0 | 3 |
| 2 | 5 | 7 |
| 3 | 2 | 6 |
| 4 | 6 | 9 |
| Moyenne | 3.25 | 6.25 |

# Shortest Job First (SJF)

## Description de l'algorithme *Shortest Job First*

Les processus les plus courts sont exécutés en premier.

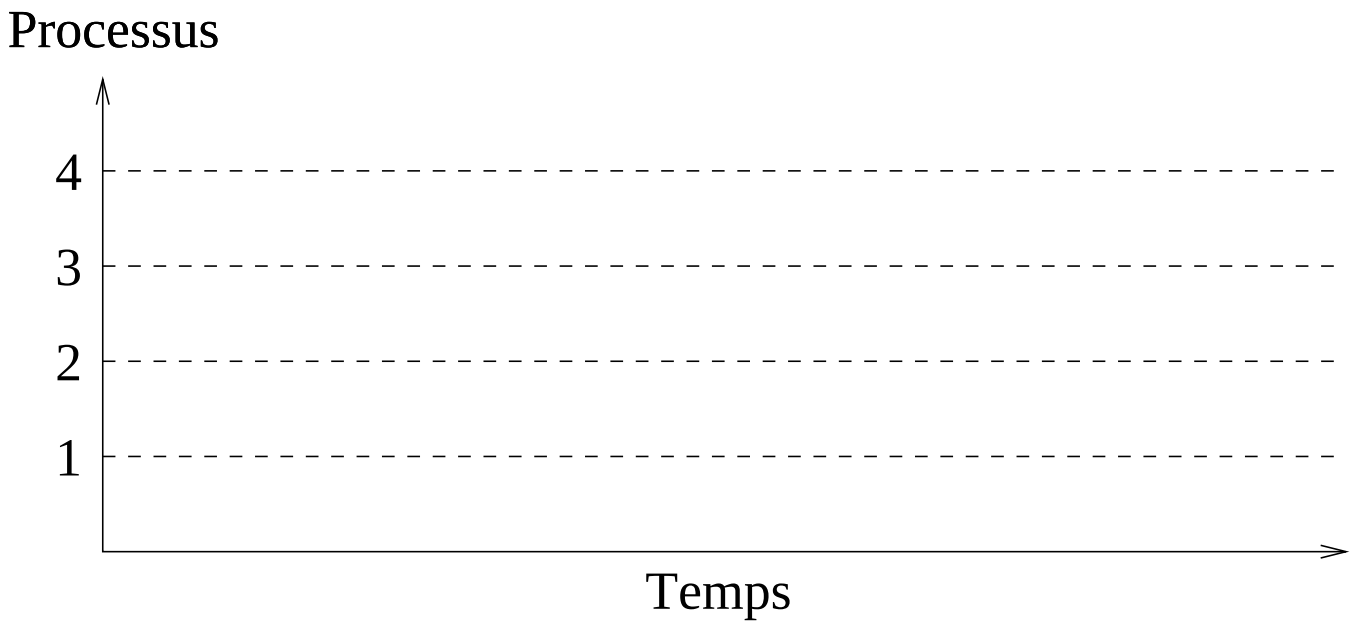
## Propriétés de l'algorithme

- Pas de notion de priorité sur les processus ;
- Non préemptif ;
- Nécessité de connaître la durée des processus ;
- Risque de privation.

# Shortest Job First : exemple

## Données des processus

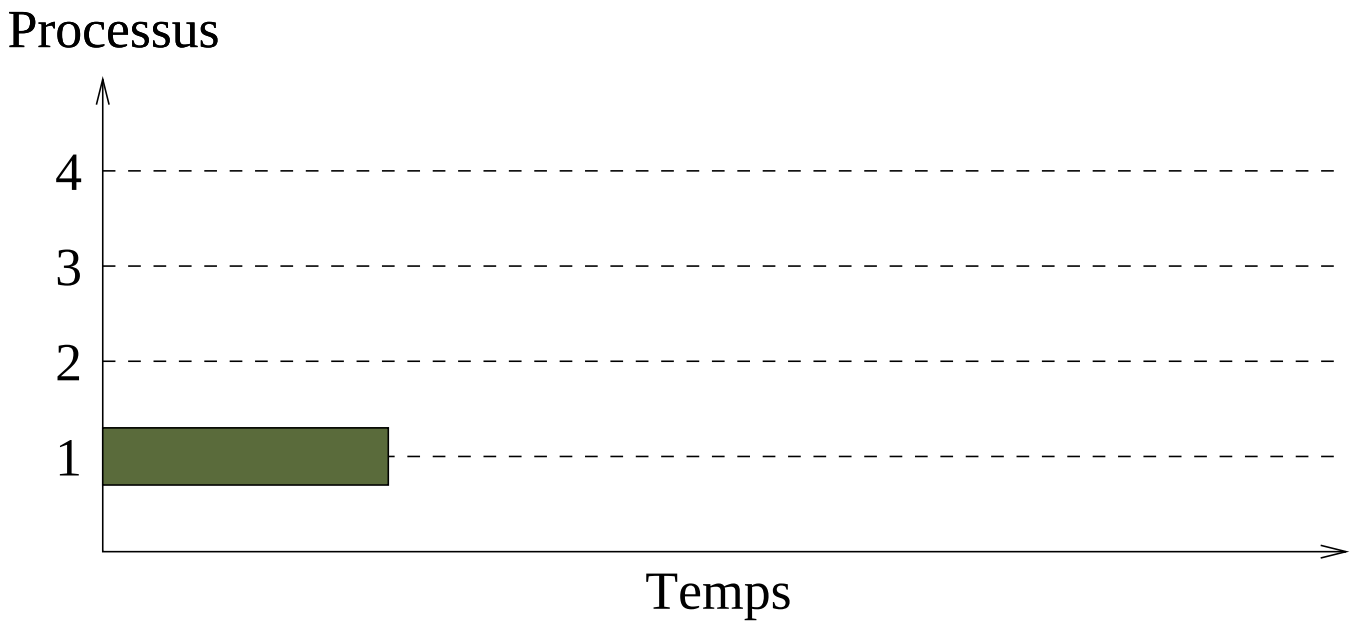
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# Shortest Job First : exemple

## Données des processus

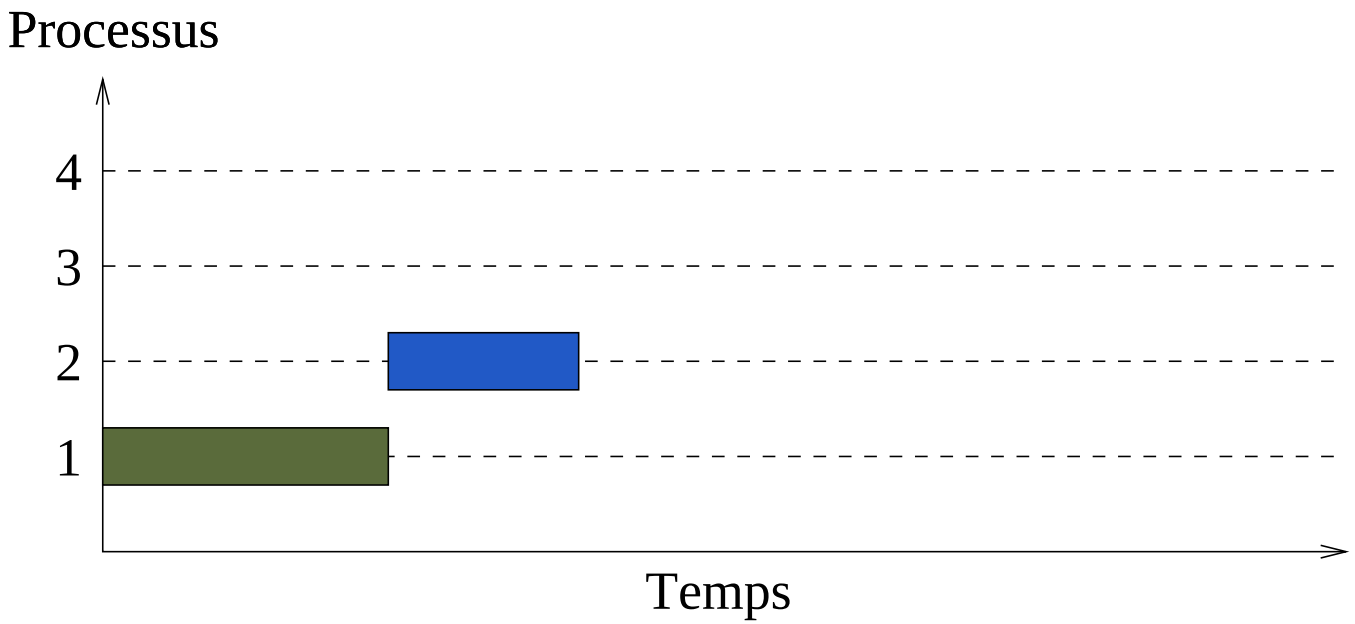
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# Shortest Job First : exemple

## Données des processus

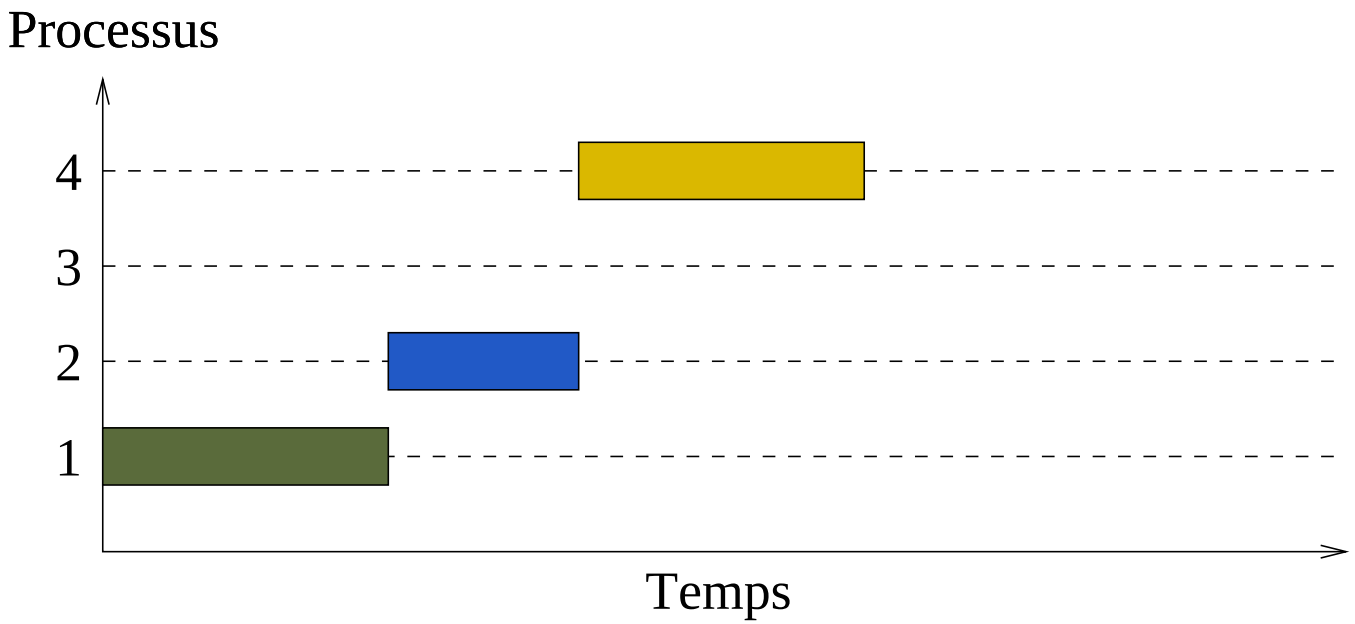
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# Shortest Job First : exemple

## Données des processus

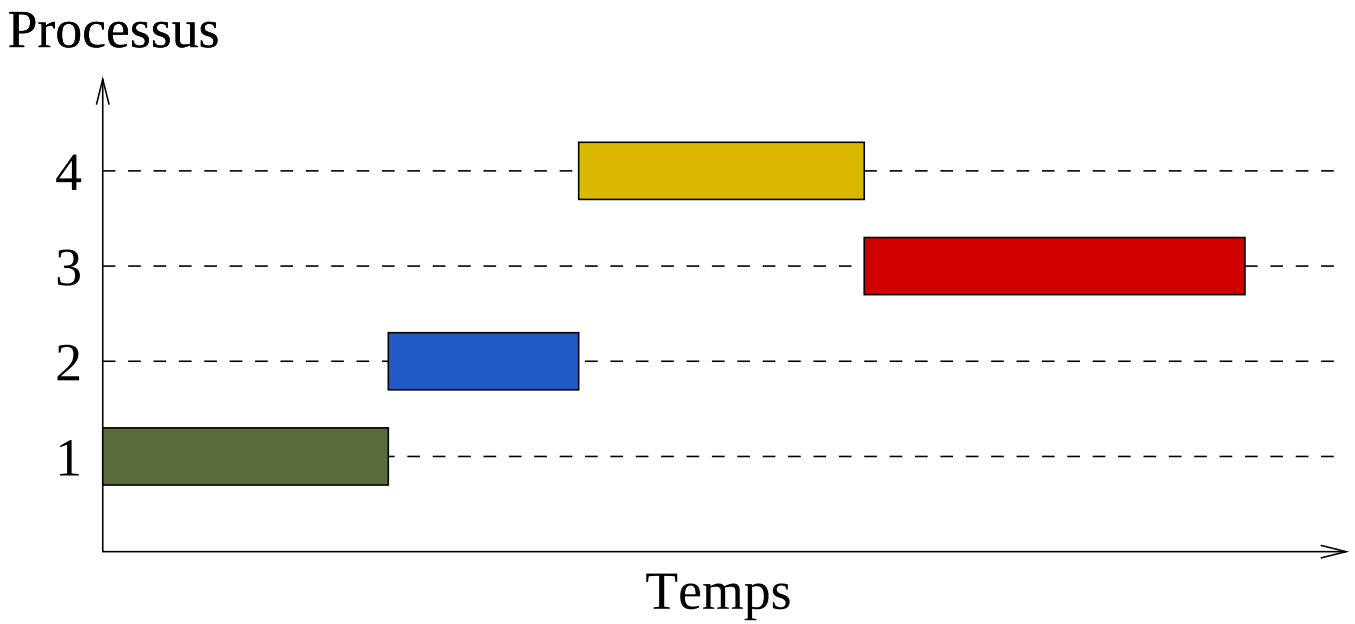
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# Shortest Job First : exemple

## Données des processus

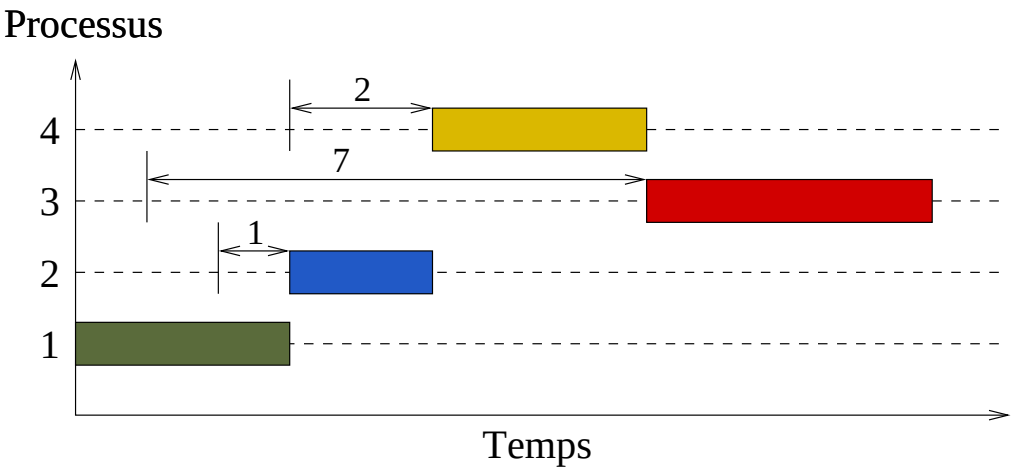
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |



# Shortest Job First : exemple

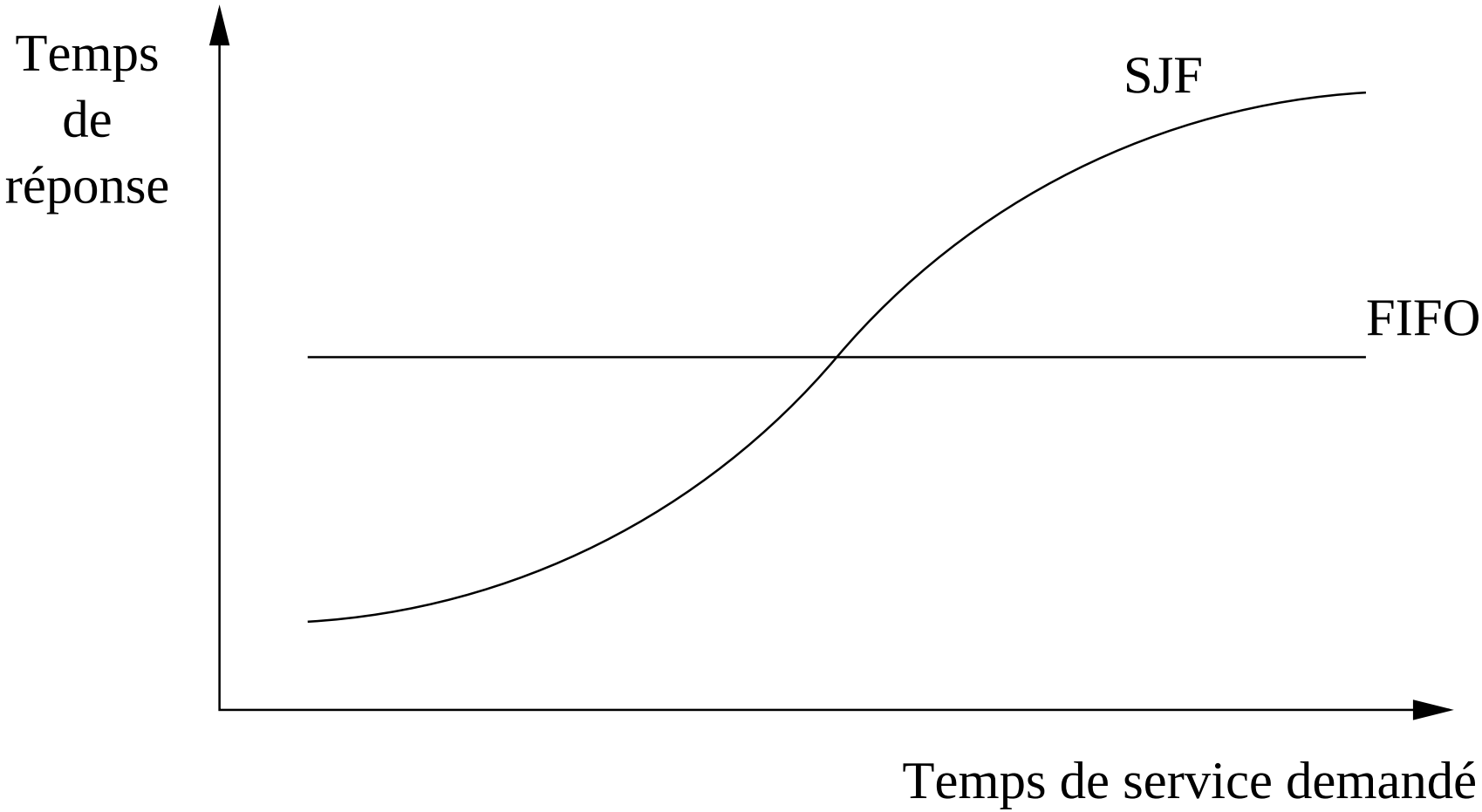
## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |




| Processus | Temps d'attente | Temps d'exécution |
|---|---|---|
| 1 | 0 | 3 |
| 2 | 1 | 3 |
| 3 | 7 | 11 |
| 4 | 2 | 5 |
| Moyenne | 2.5 | 5.5 |

# Temps de réponse : FIFO versus SJF



# Shortest Remaining Time (SRT)

## Description de l'algorithme *Shortest Remaining Time*

Les processus dont il reste le temps d'exécution le plus court sont exécutés en premier.

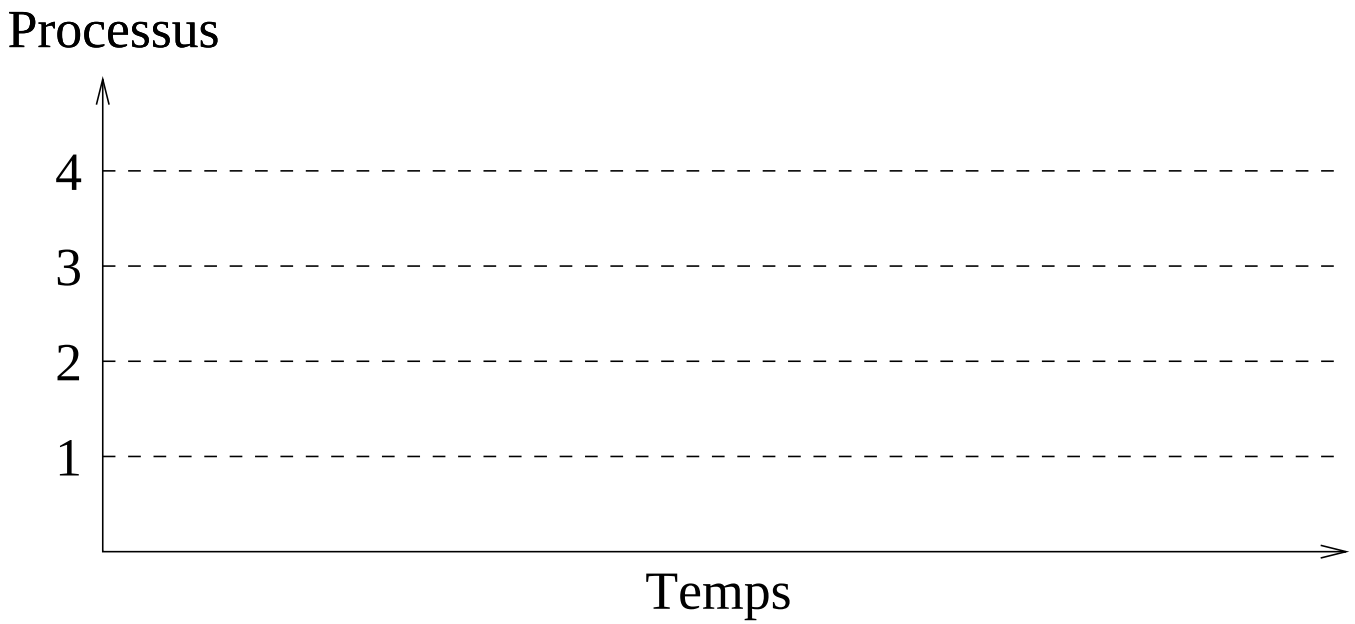
## Propriétés de l'algorithme

- Pas de notion de priorité sur les processus ;
- Préemptif : par l'arrivée d'un nouveau processus ;
- Nécessité de connaître la durée des processus ;
- On favorise les travaux courts ;
- Coûts d'exploitation importants.

# Shortest Remaining Time Next : exemple

## Données des processus

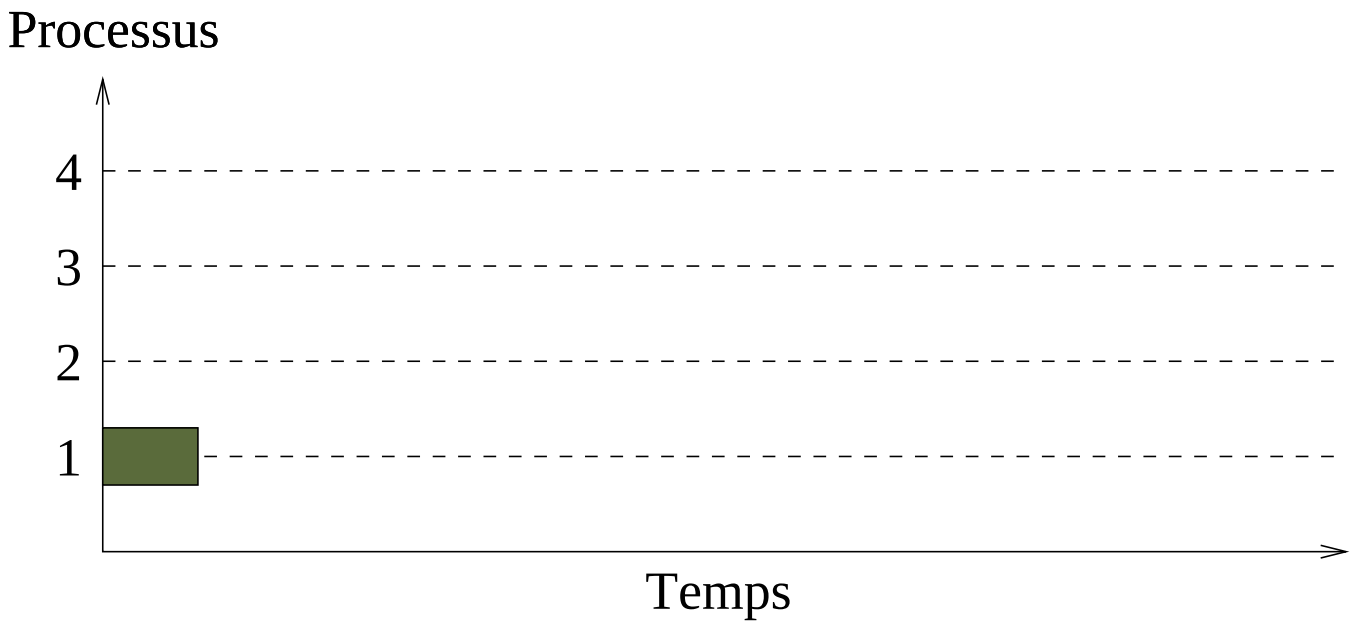
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |



# Shortest Remaining Time Next : exemple

## Données des processus

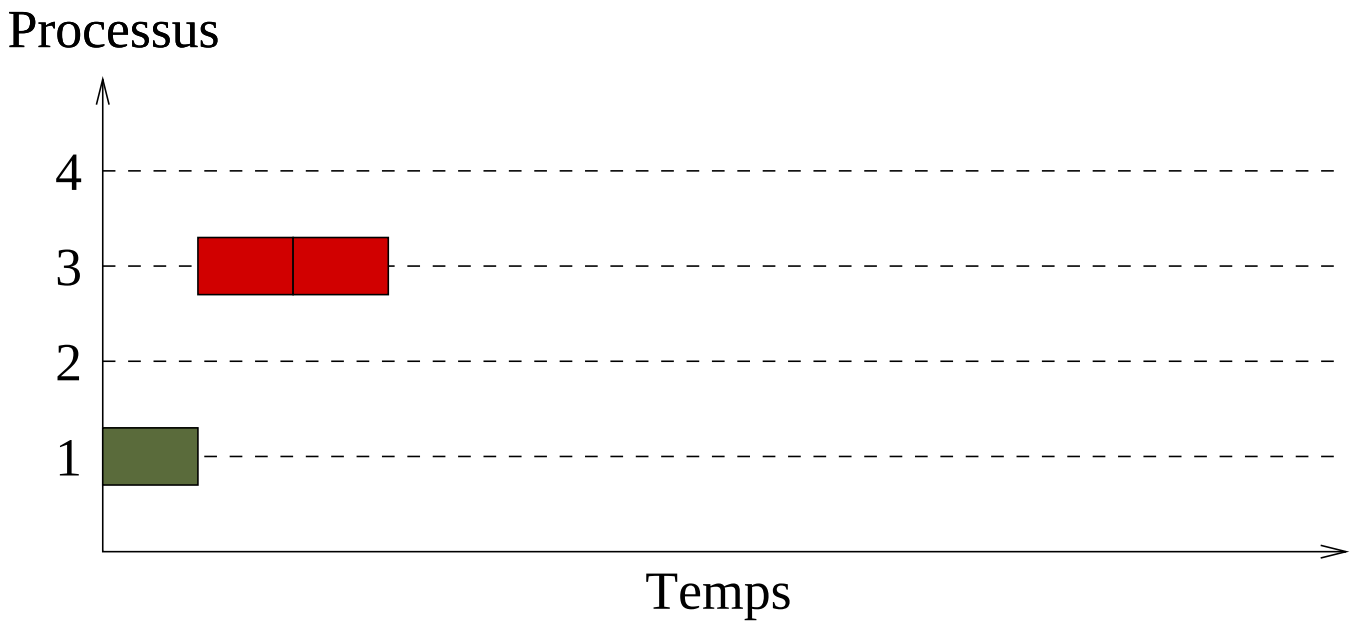
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |



# Shortest Remaining Time Next : exemple

## Données des processus

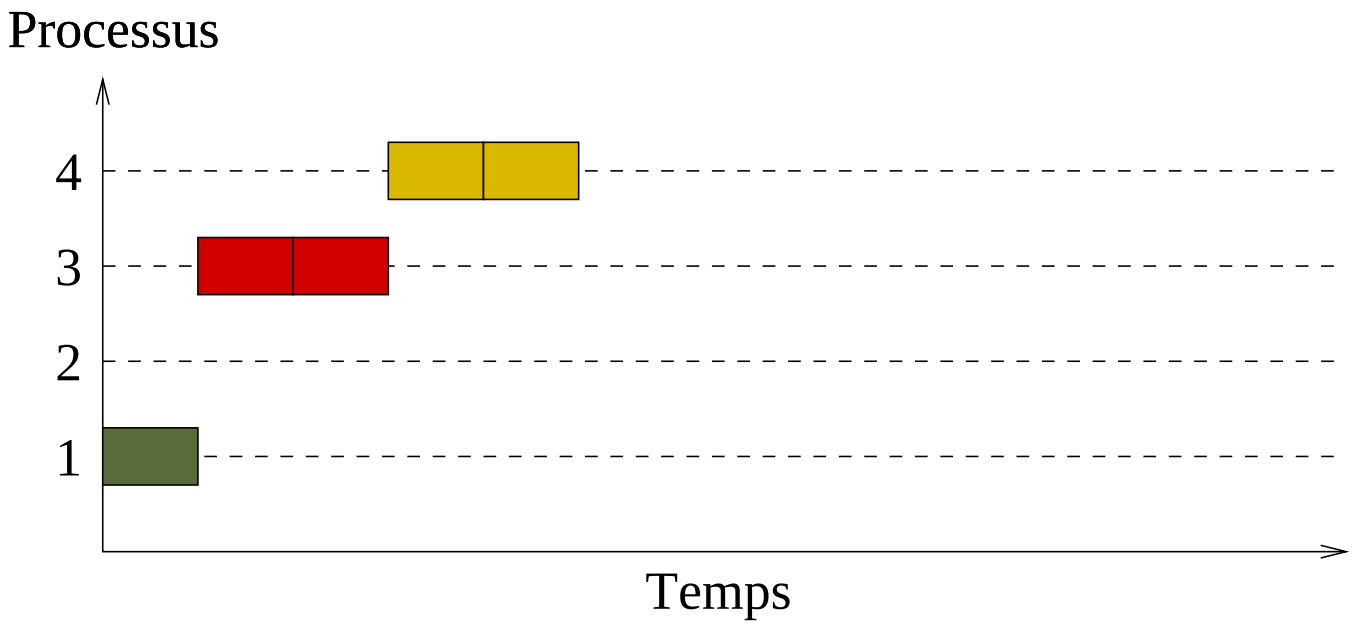
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |



# Shortest Remaining Time Next : exemple

## Données des processus

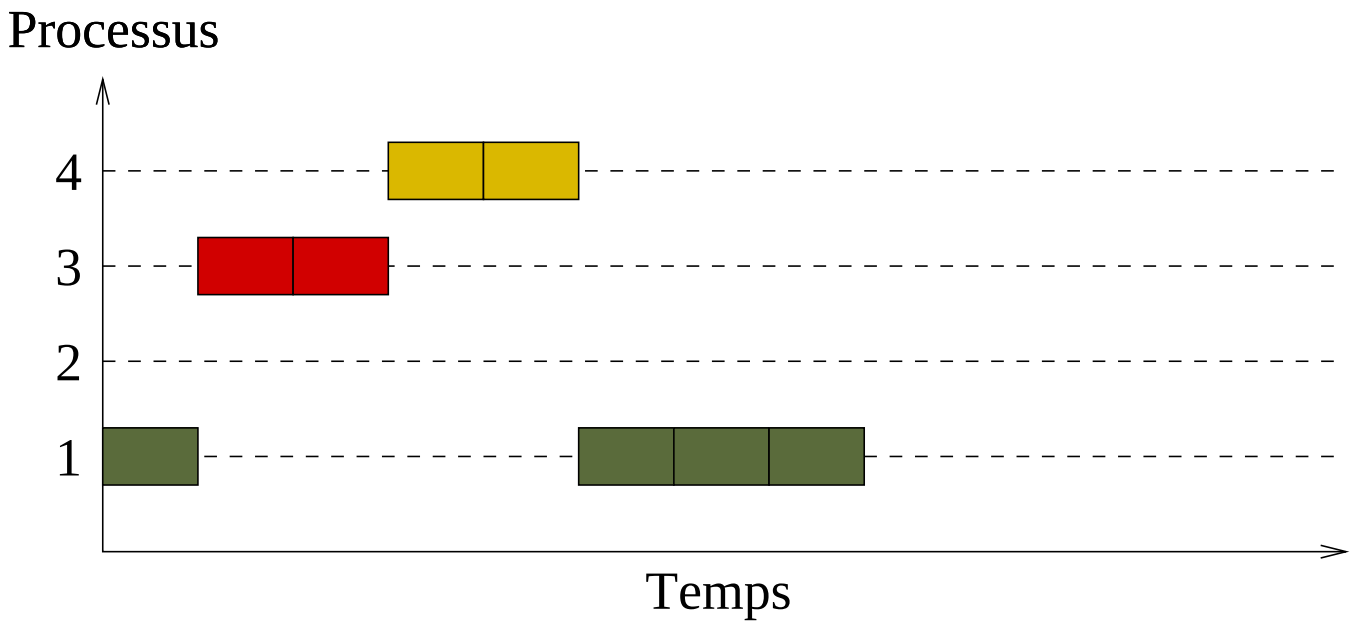
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |



# Shortest Remaining Time Next : exemple

## Données des processus

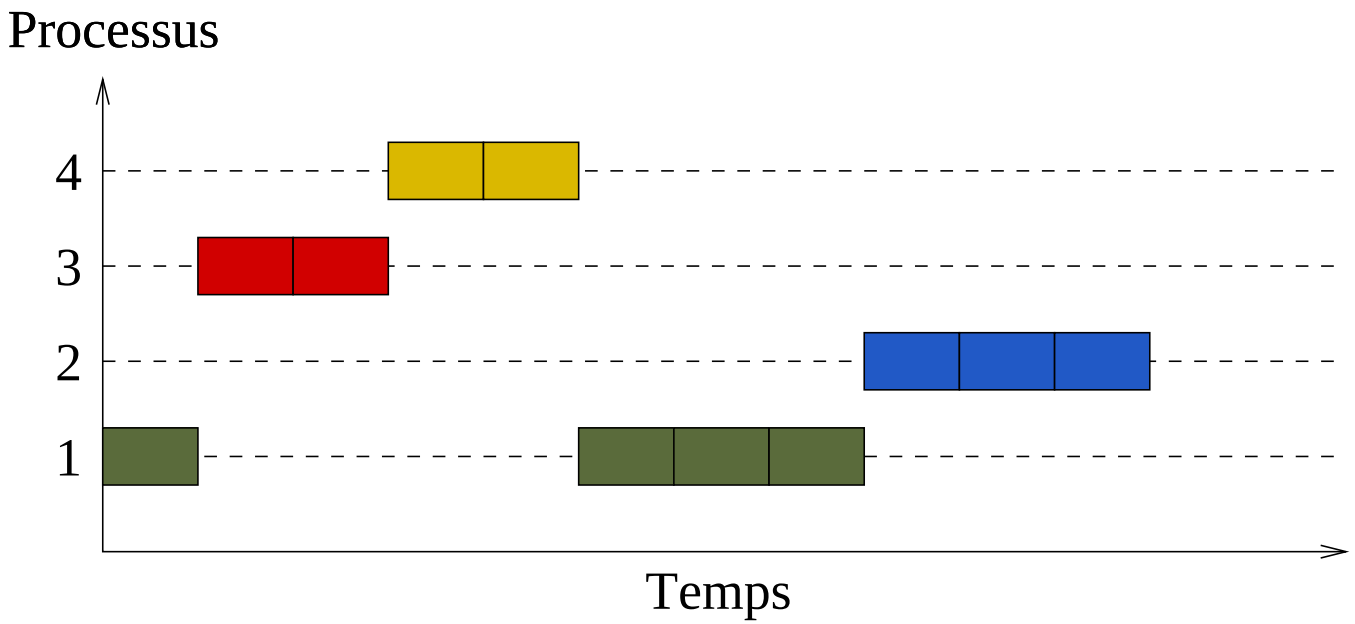
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |



# Shortest Remaining Time Next : exemple

## Données des processus

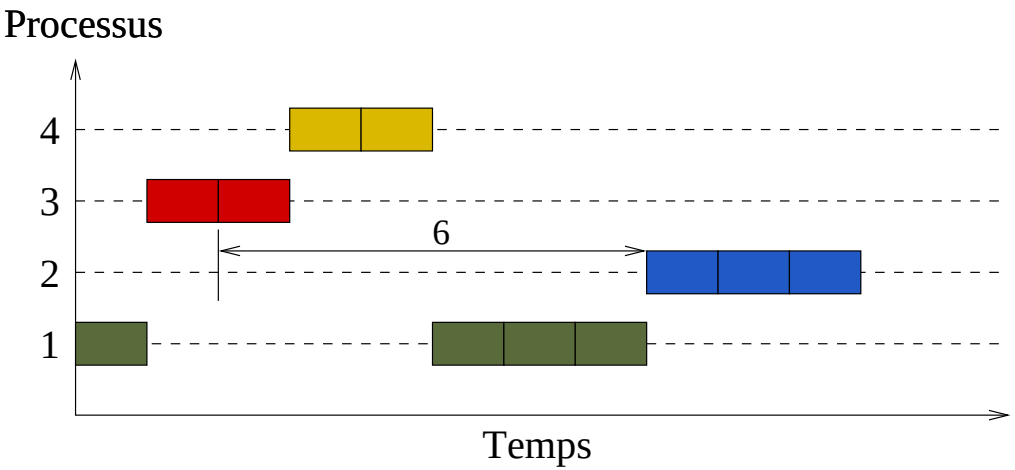
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |



# Shortest Remaining Time Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 4 | 0 |
| 2 | 3 | 2 |
| 3 | 2 | 1 |
| 4 | 2 | 3 |




| Processus | Temps d'attente | Temps d'exécution |
|---|---|---|
| 1 | 0 | 8 |
| 2 | 6 | 9 |
| 3 | 0 | 2 |
| 4 | 0 | 2 |
| Moyenne | 1.5 | 5.25 |

# Highest Response Ratio Next (HRN)

## Description de l'algorithme *Highest Response Ratio Next*

Le choix du processus est basé sur le ratio :

$$p(t) = \frac{w(t) + t_s}{t_s}$$

avec

- $w(t)$ : temps d'attente,
- $t_s$ : temps d'exécution estimé.

Si $w(t) = 0$, les travaux les plus courts ($t_s$) sont privilégiés.

## Propriétés de l'algorithme

- Pas de notion de priorité sur les processus ;
- Préemptif et non-préemptif ;
- Nécessité de connaître la durée des processus ;
- On favorise les travaux courts ou qui attend à long temps ;

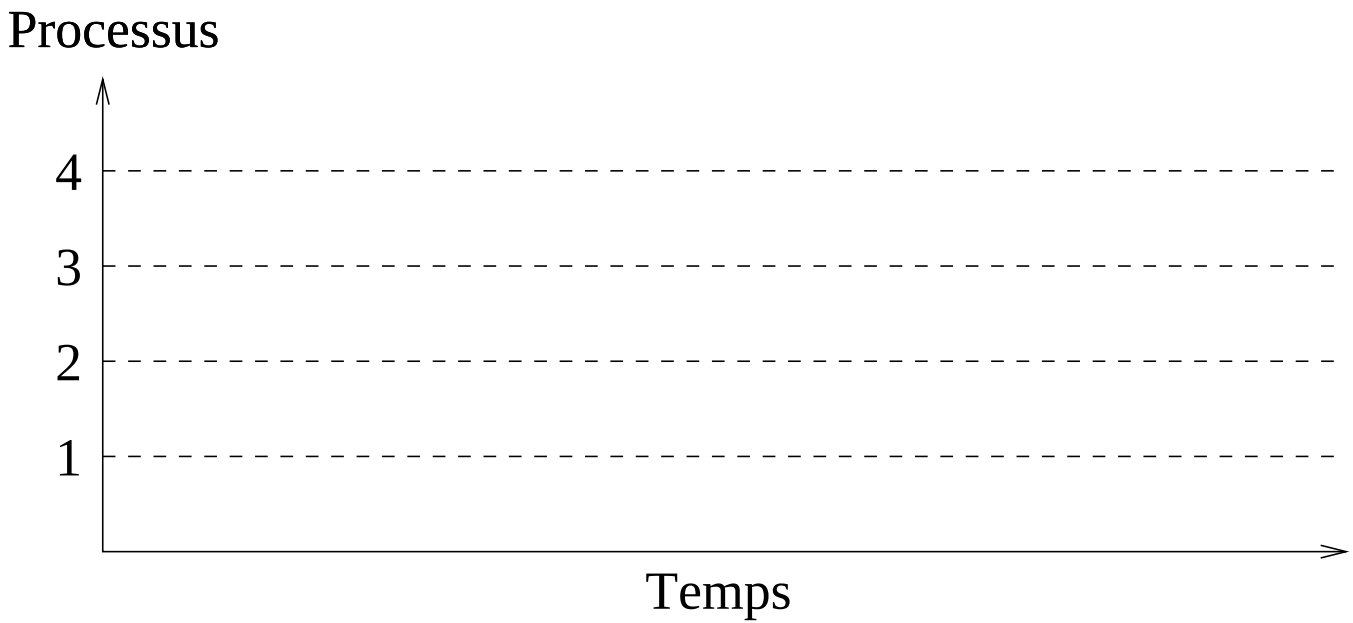
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | | | | | | | | | | | |
| 4 | | | | | | | | | | | | |



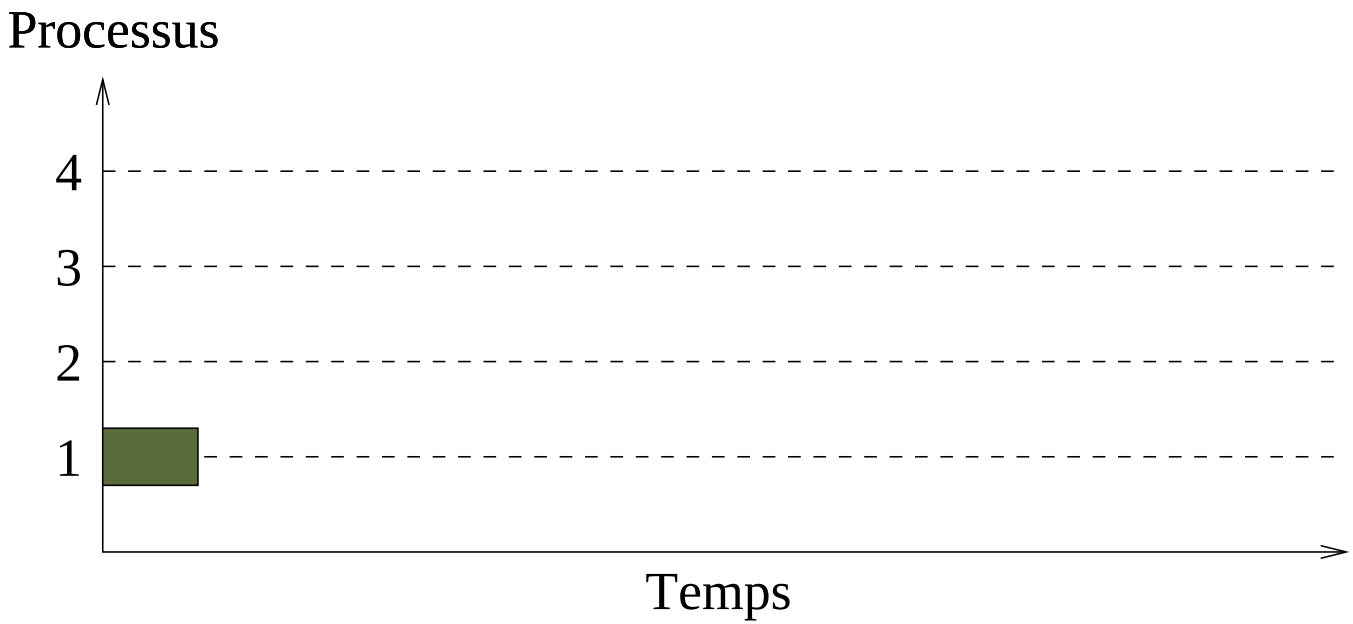
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | | $\frac{4}{4}$ | | | | | | | | | | |
| 4 | | | | | | | | | | | | |



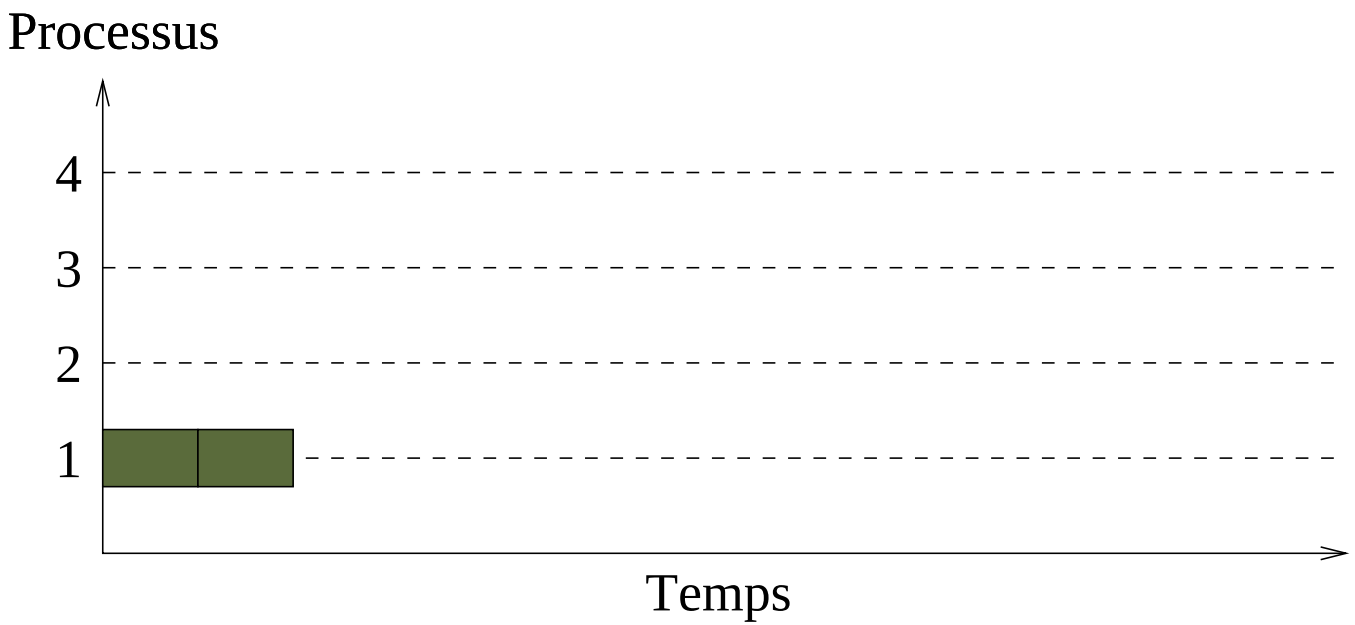
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | | | | | | | | | |
| 2 | | | $\frac{2}{2}$ | | | | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | | | | | | | | | |
| 4 | | | | | | | | | | | | |



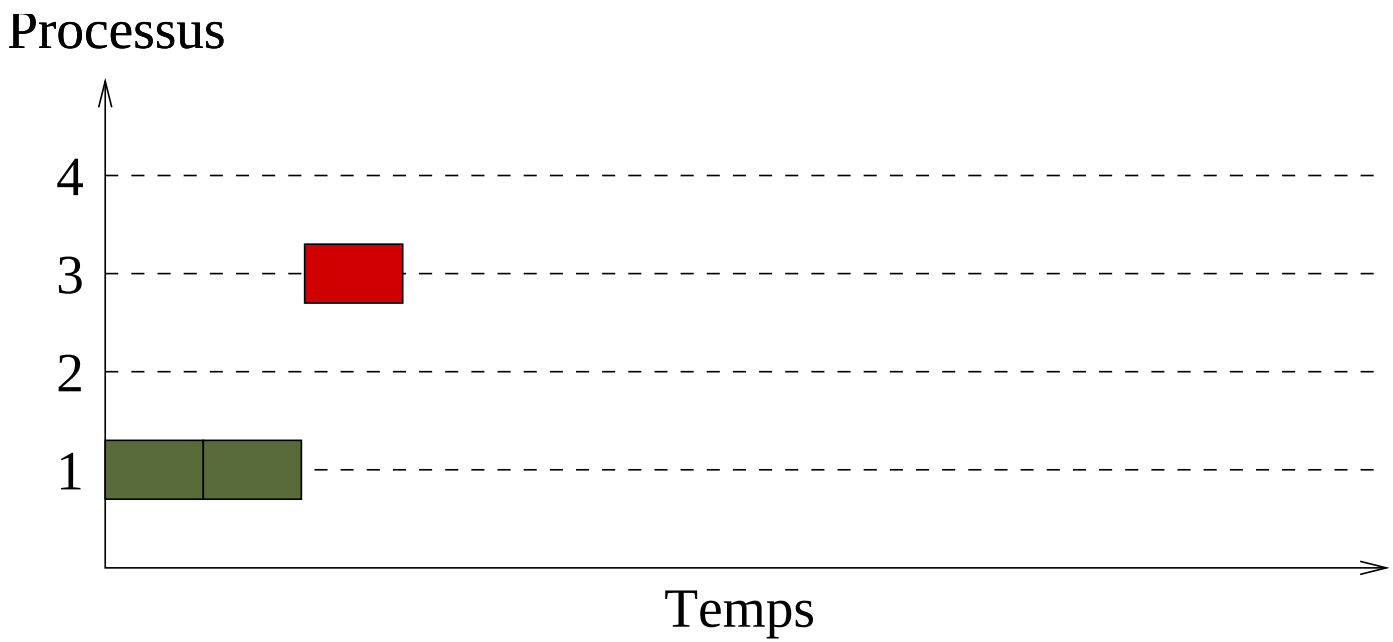
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | | | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | | | | | | | | |
| 4 | | | | $\frac{3}{3}$ | | | | | | | | |



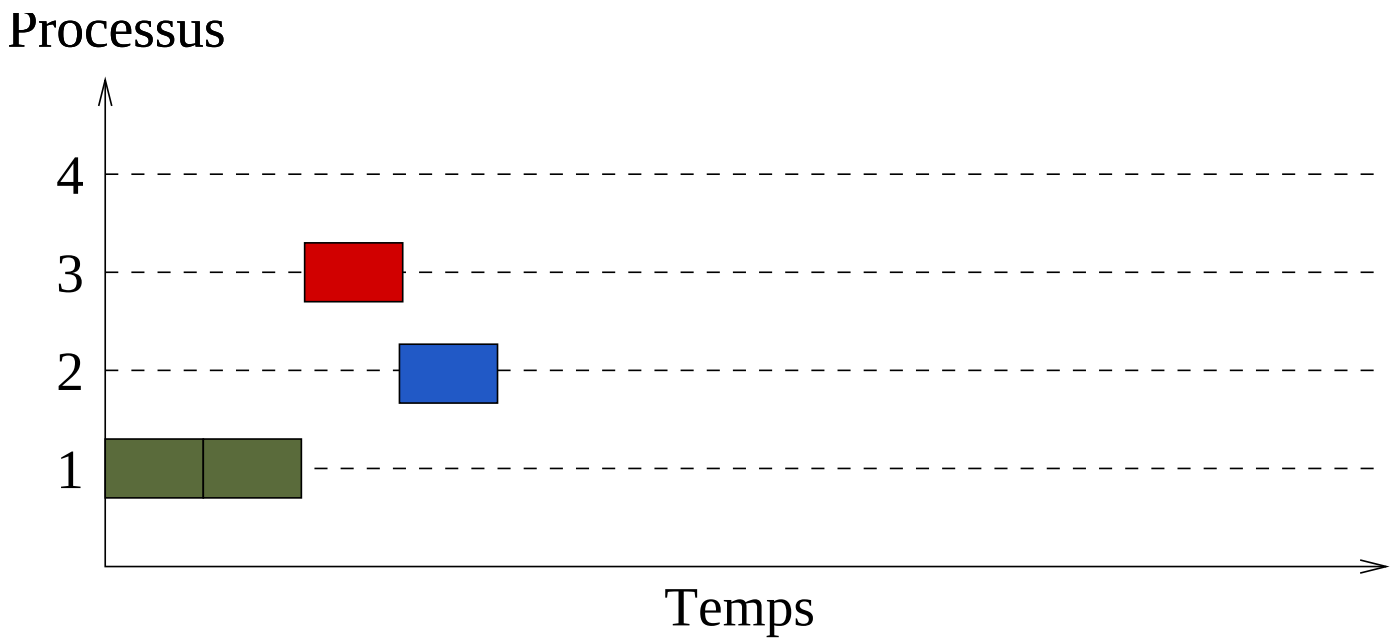
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | | | | | | | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | | | | | | | |



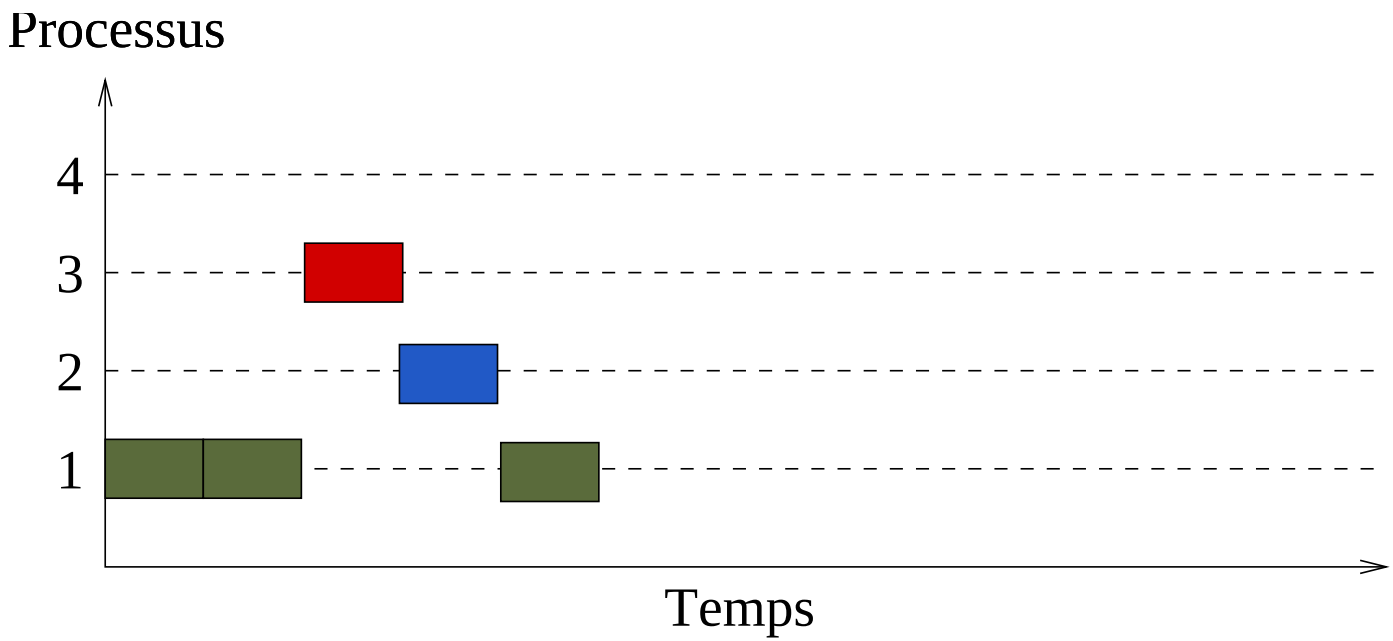
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | | | | | | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | |



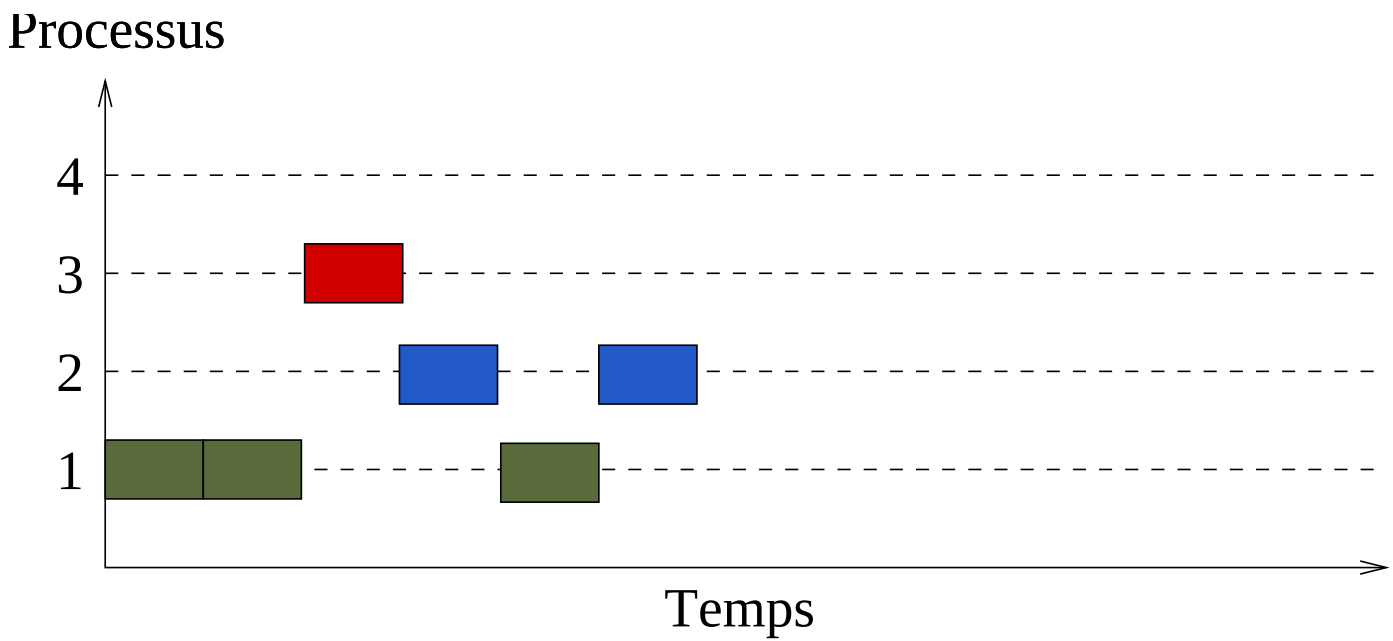
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | | | | | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | | | | | |



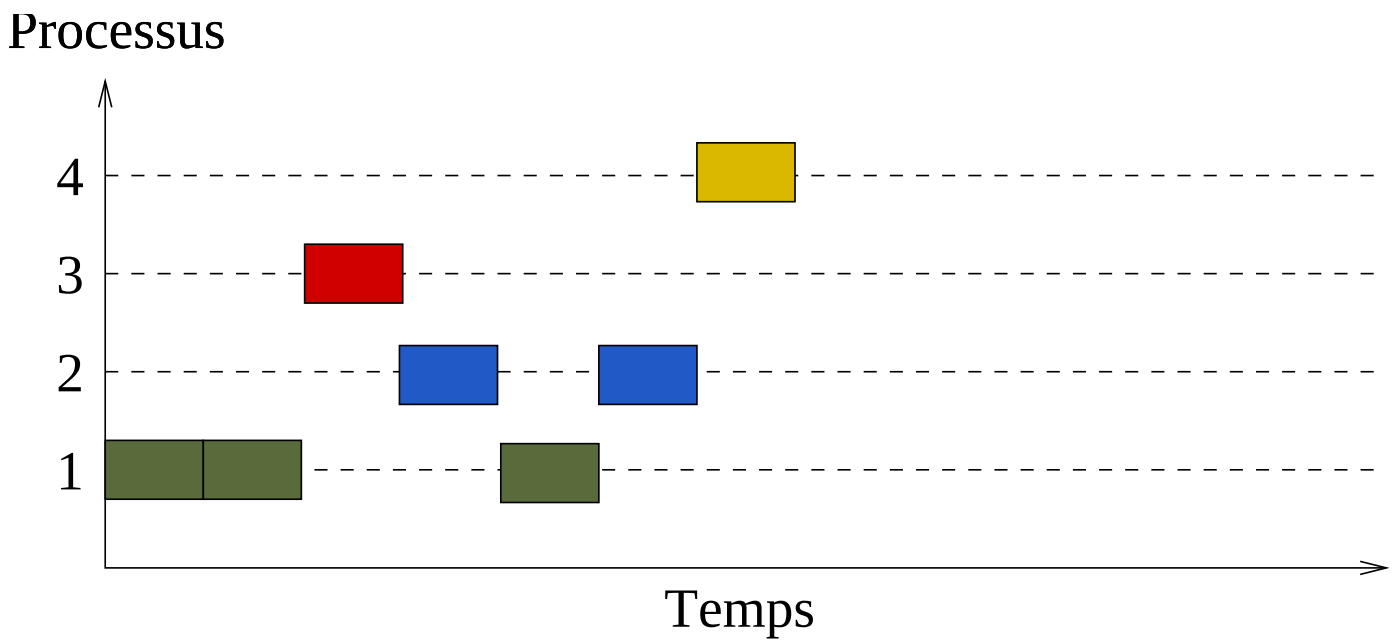
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | | | | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | | | | |



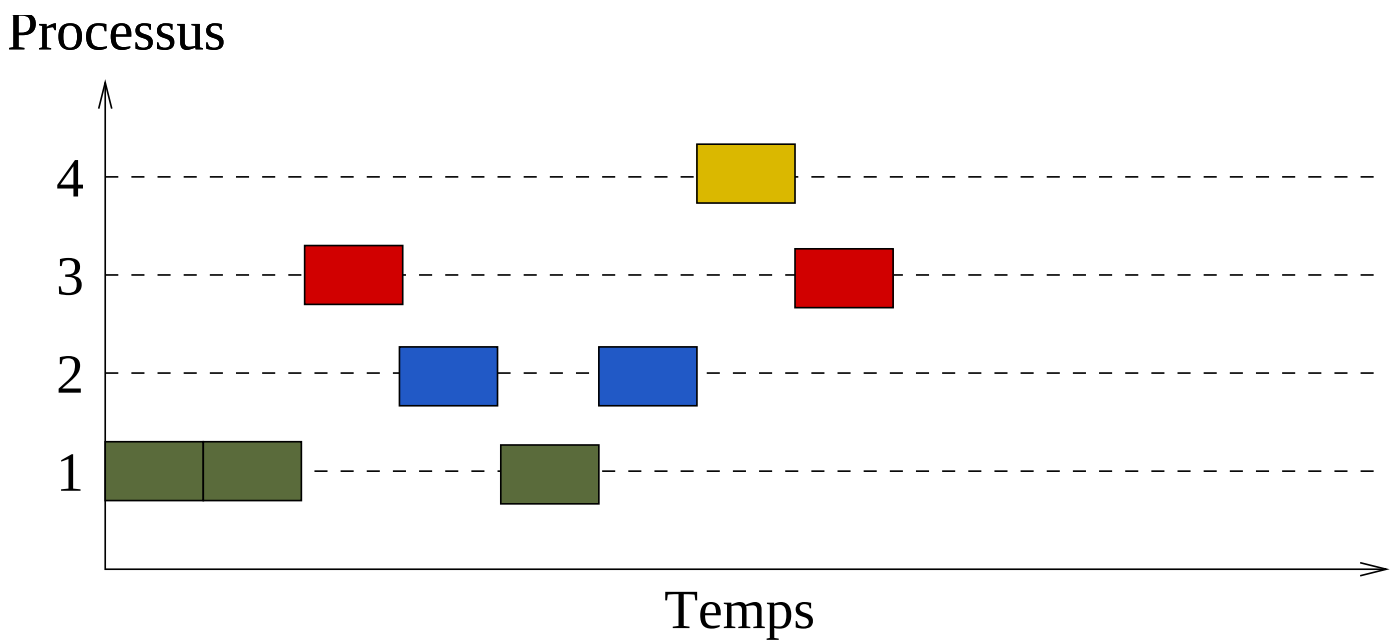
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | $\frac{9}{4}$ | | | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | $\frac{7}{3}$ | | | |



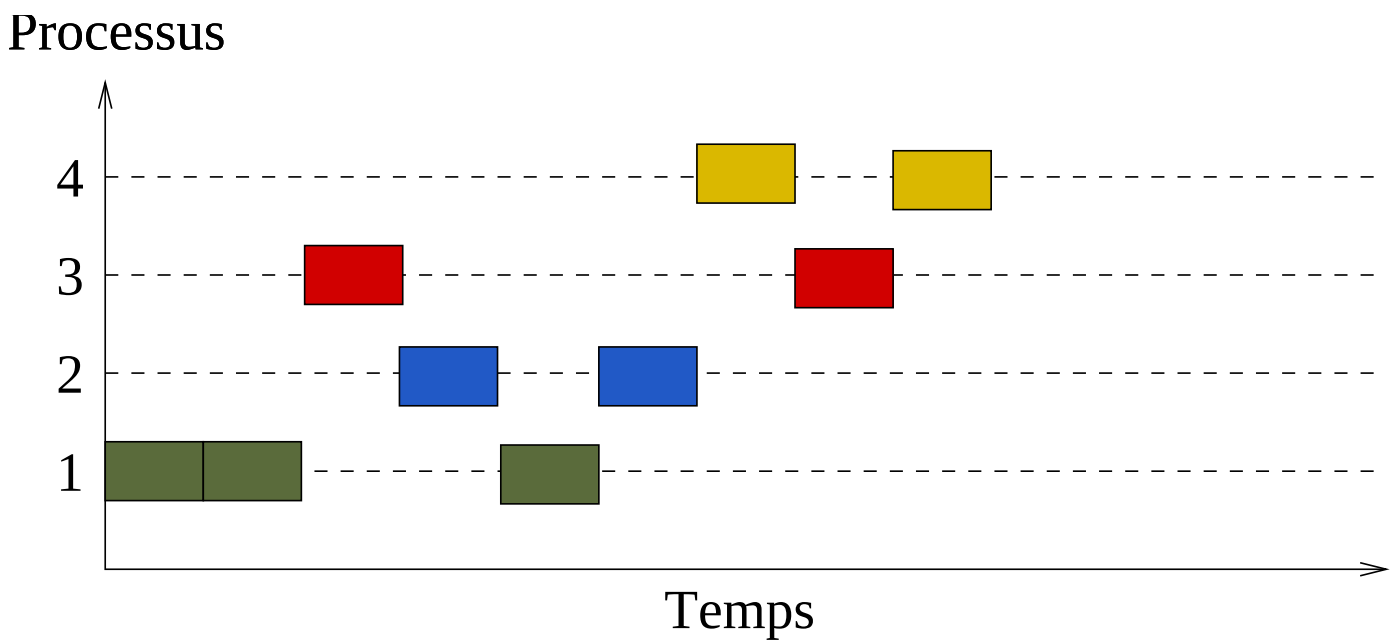
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | $\frac{9}{4}$ | $\frac{10}{4}$ | | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | $\frac{7}{3}$ | $\frac{7}{3}$ | | |



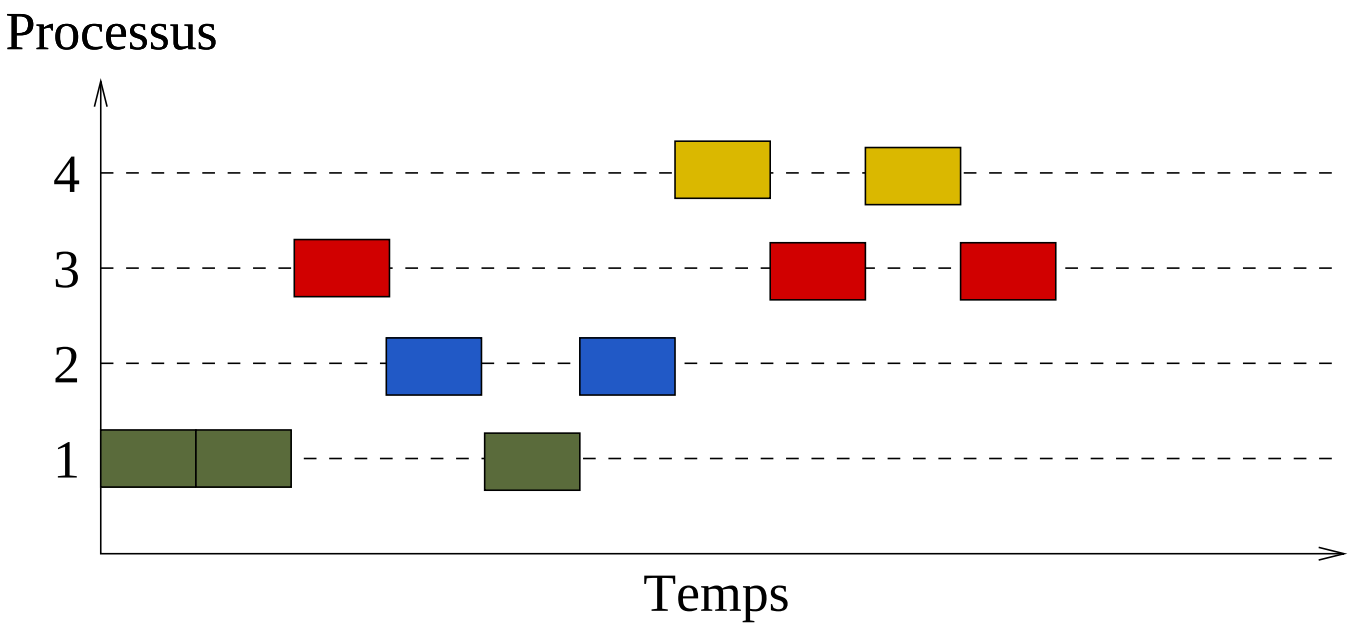
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | $\frac{9}{4}$ | $\frac{10}{4}$ | $\frac{10}{4}$ | |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | $\frac{7}{3}$ | $\frac{7}{3}$ | $\frac{8}{3}$ | |



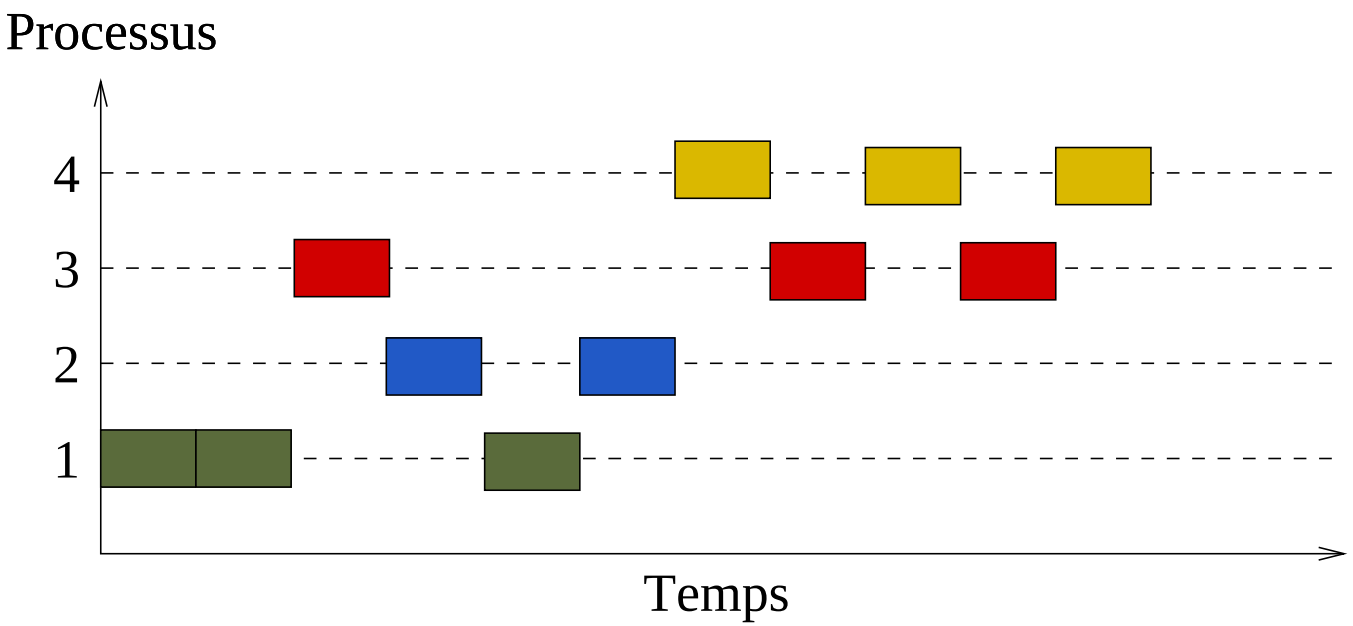
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | $\frac{9}{4}$ | $\frac{10}{4}$ | $\frac{10}{4}$ | $\frac{11}{4}$ |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | $\frac{7}{3}$ | $\frac{7}{3}$ | $\frac{8}{3}$ | |



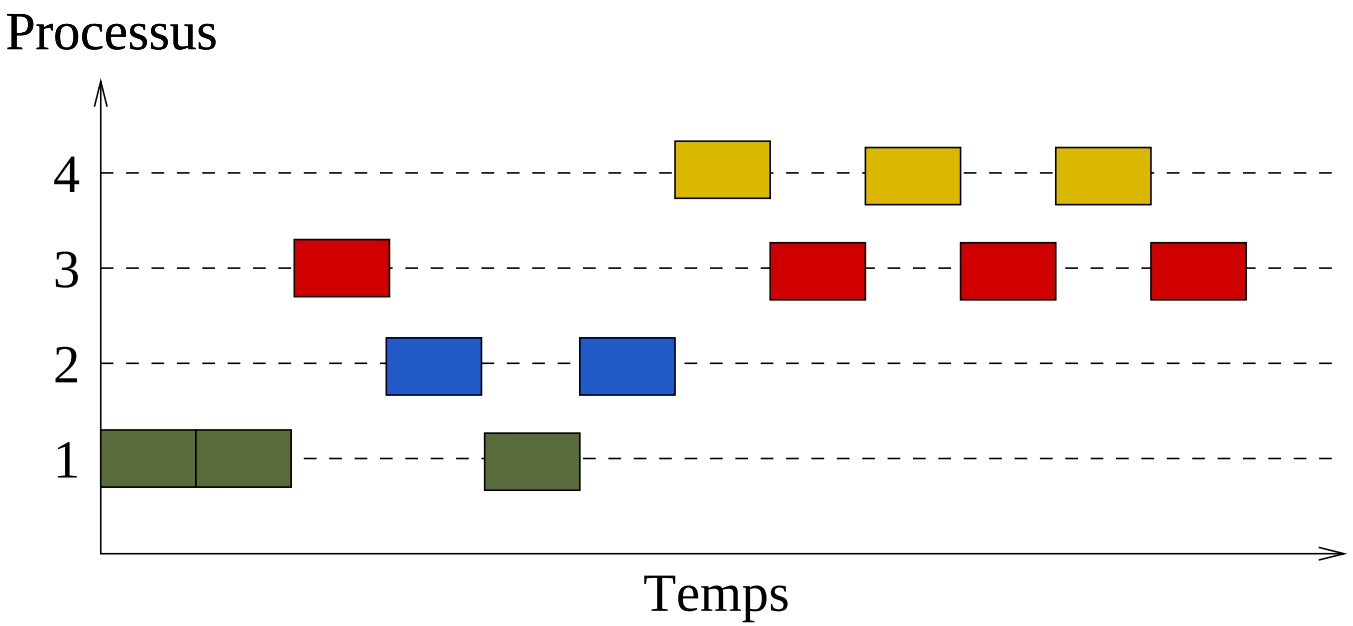
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | $\frac{9}{4}$ | $\frac{10}{4}$ | $\frac{10}{4}$ | $\frac{11}{4}$ |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | $\frac{7}{3}$ | $\frac{7}{3}$ | $\frac{8}{3}$ | |



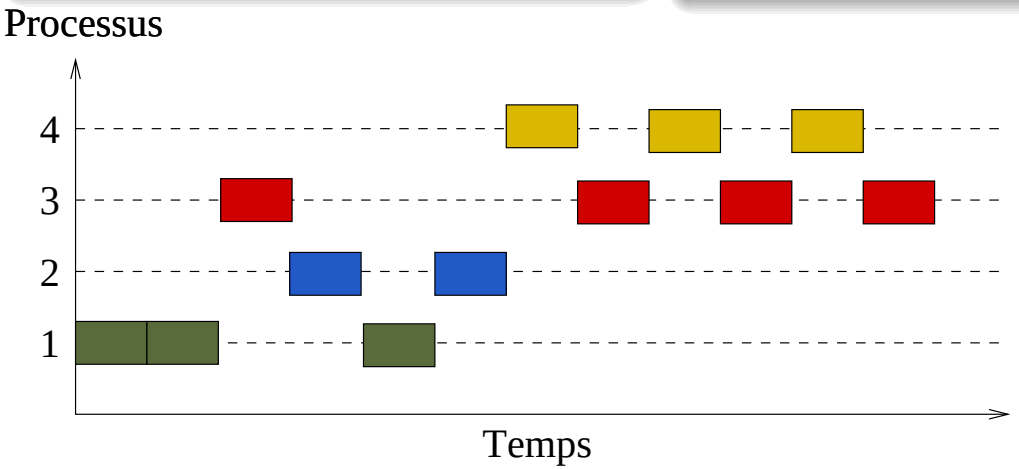
# Highest Response Ratio Next : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

## Calcul des priorités

| Processus | Temps | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 1 | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | | | | | | | |
| 2 | | | $\frac{2}{2}$ | $\frac{3}{2}$ | $\frac{3}{2}$ | $\frac{4}{2}$ | | | | | | |
| 3 | | $\frac{4}{4}$ | $\frac{5}{4}$ | $\frac{5}{4}$ | $\frac{6}{4}$ | $\frac{7}{4}$ | $\frac{8}{4}$ | $\frac{9}{4}$ | $\frac{9}{4}$ | $\frac{10}{4}$ | $\frac{10}{4}$ | $\frac{11}{4}$ |
| 4 | | | | $\frac{3}{3}$ | $\frac{4}{3}$ | $\frac{5}{3}$ | $\frac{6}{3}$ | $\frac{6}{3}$ | $\frac{7}{3}$ | $\frac{7}{3}$ | $\frac{8}{3}$ | |




| Processus | Temps d'attente | Temps d'exécution |
|---|---|---|
| 1 | 0 | 5 |
| 2 | 1 | 4 |
| 3 | 1 | 11 |
| 4 | 3 | 8 |
| Moyenne | 1.25 | 7 |

# Round Robin (Tourniquet)

## Description de l'algorithme *Round Robin*

Les processus sont rangés dans une file. Un intervalle de temps, appelé *quantum*, est assigné à chaque job. Si l'exécution du processus n'est pas terminée à la fin du quantum, il est replacé en fin de file.

## Propriétés de l'algorithme

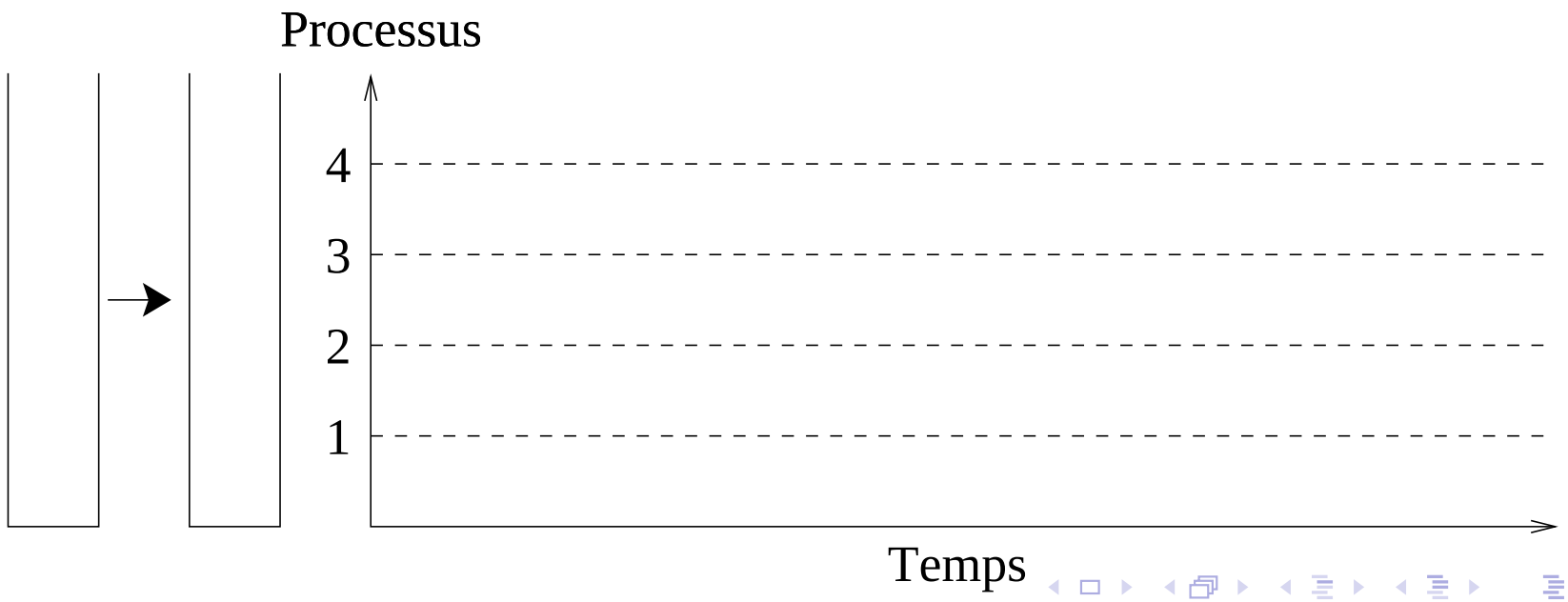
- Pas de notion de priorité sur les processus ;
- Préemptif ;
- Pas de connaissance sur la durée des processus.

# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

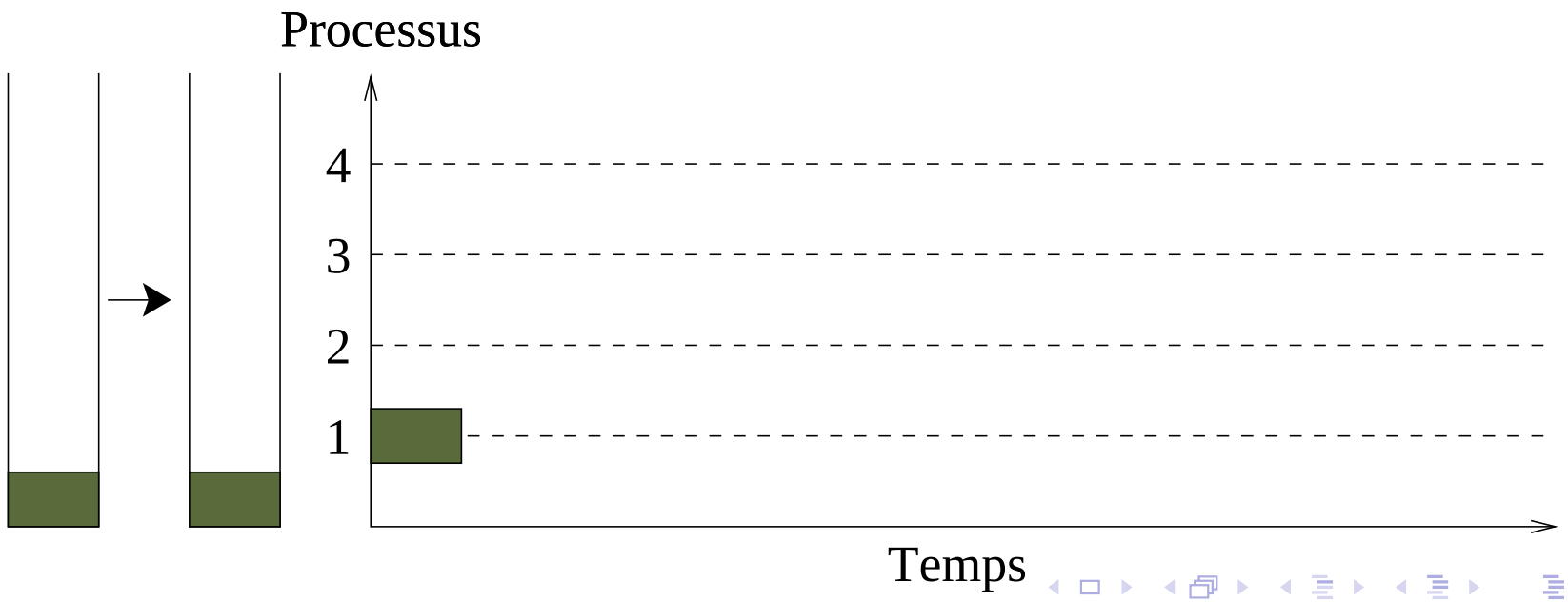


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

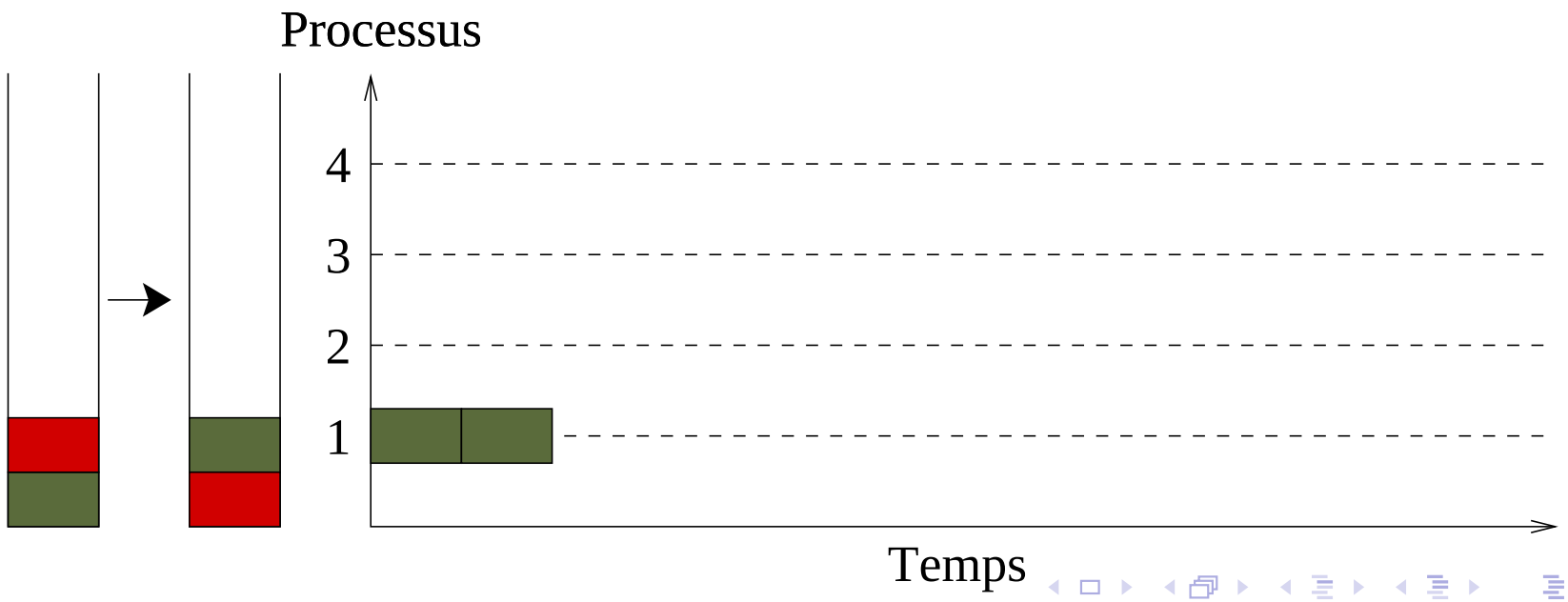


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

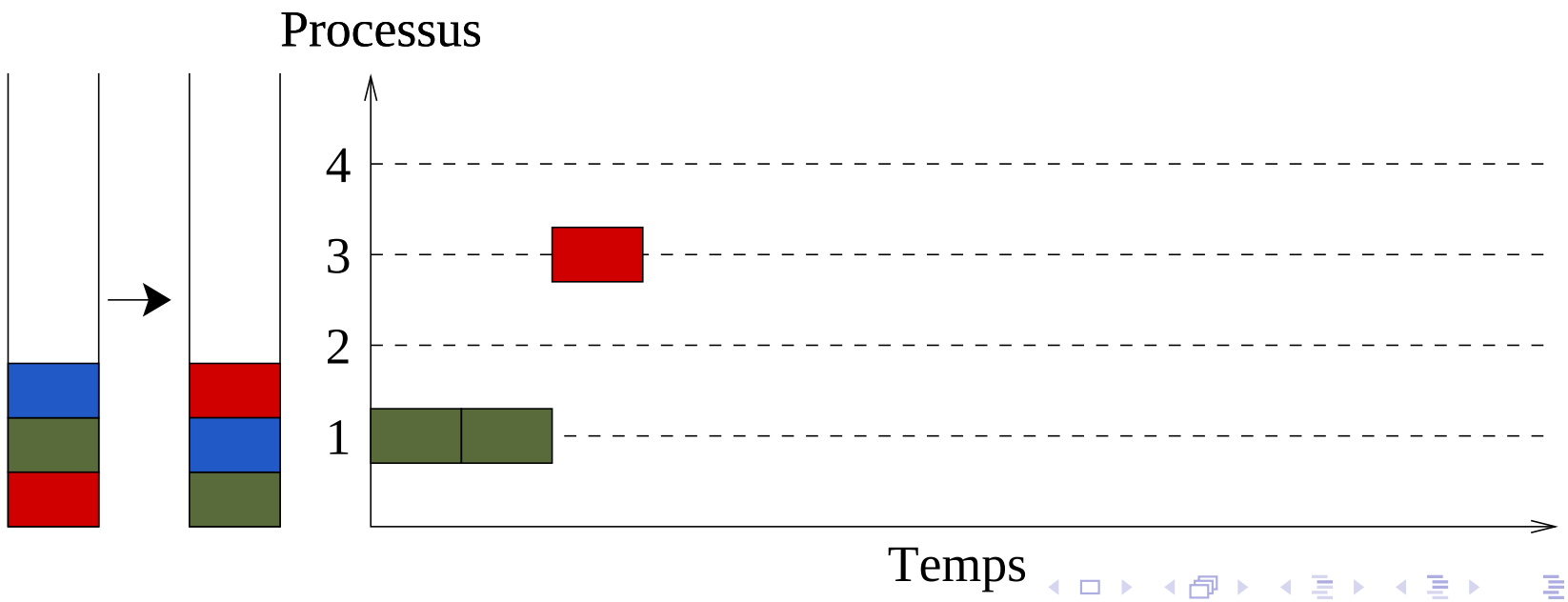


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

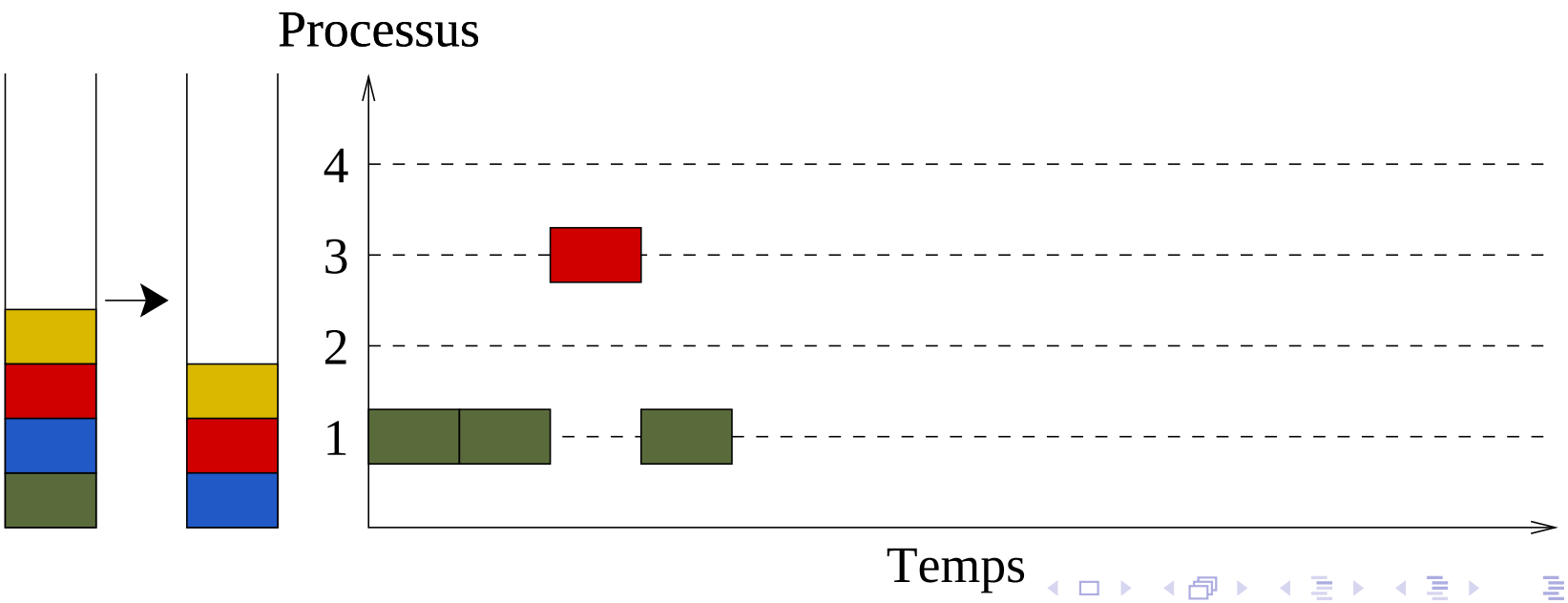


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

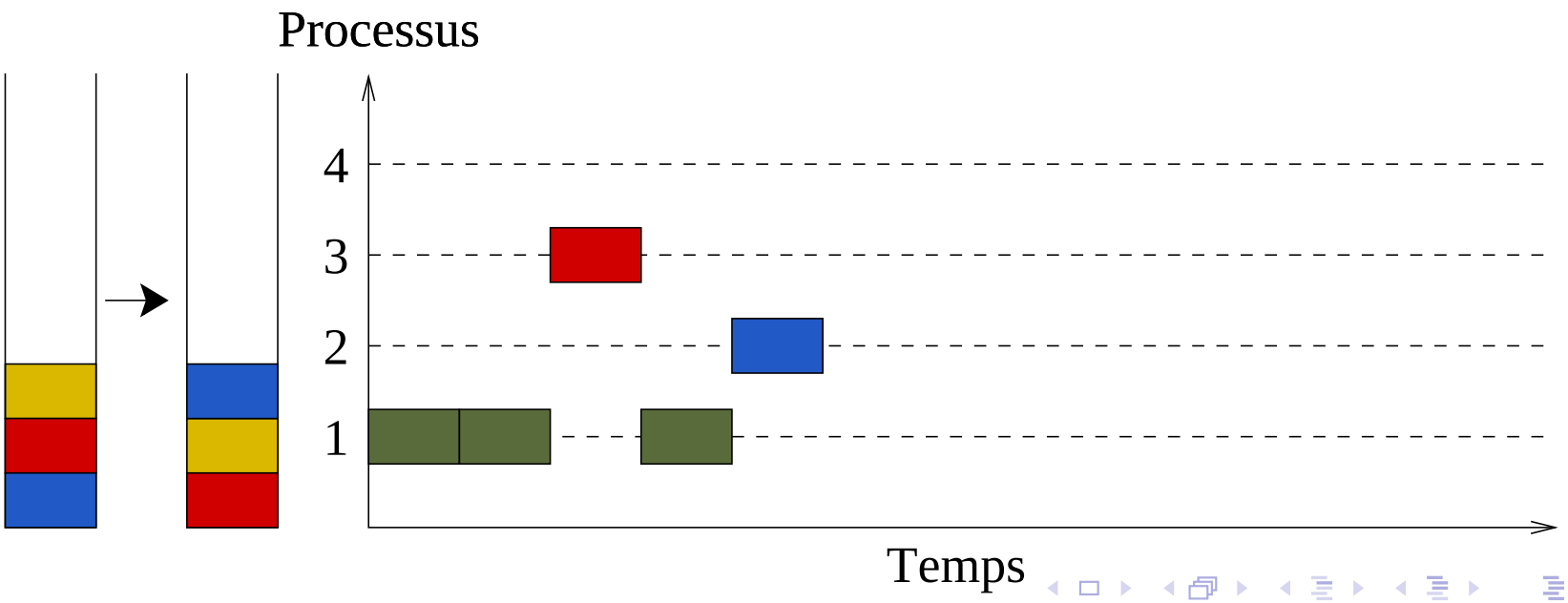


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

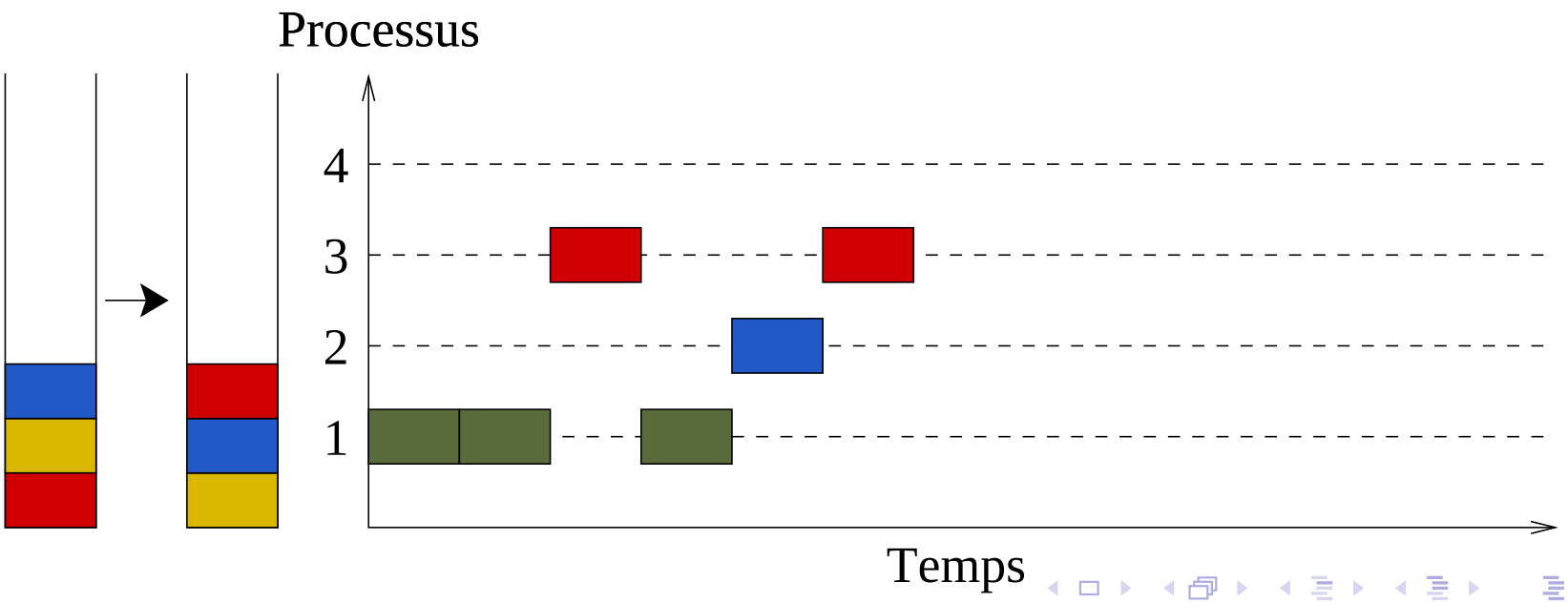


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

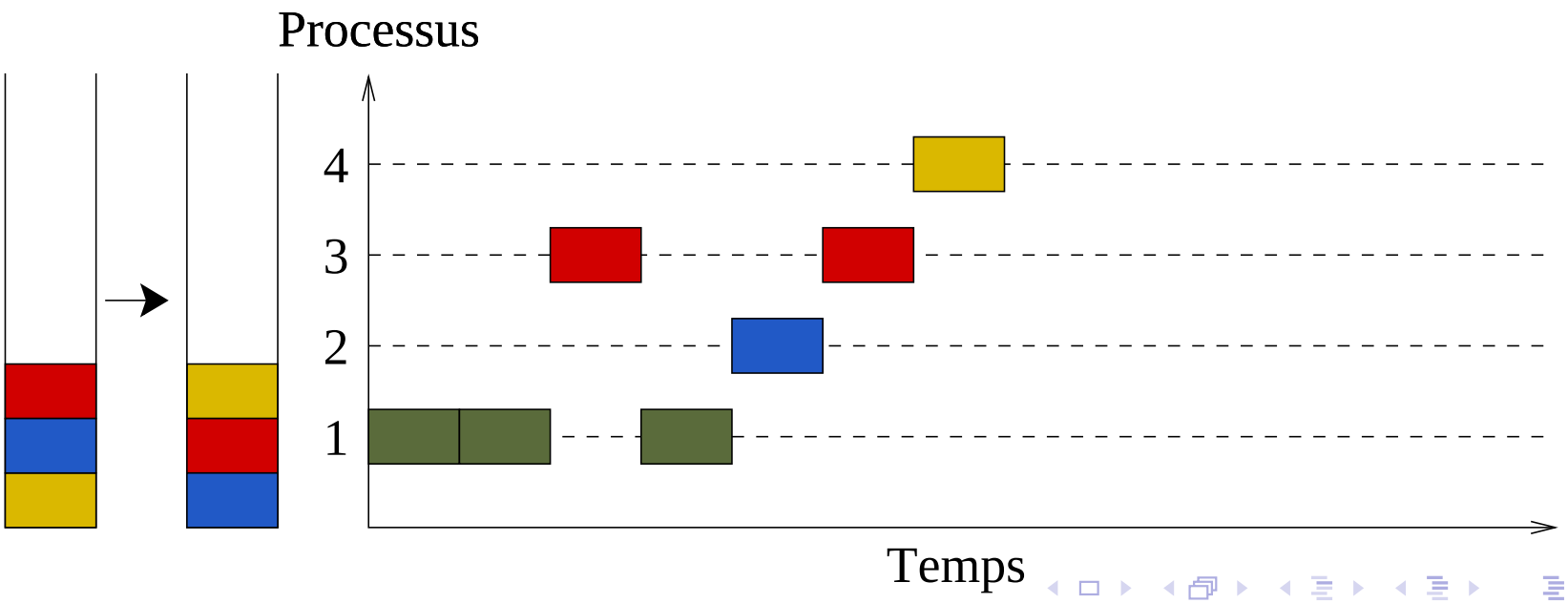


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

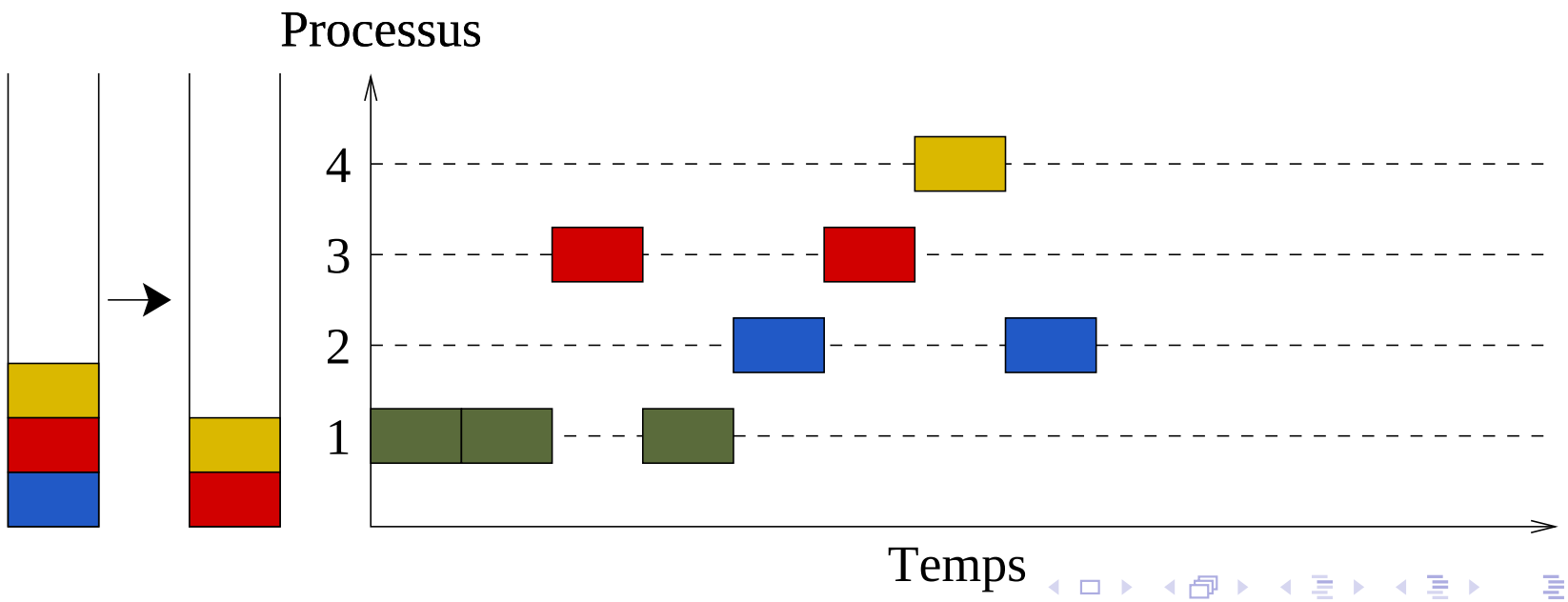


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

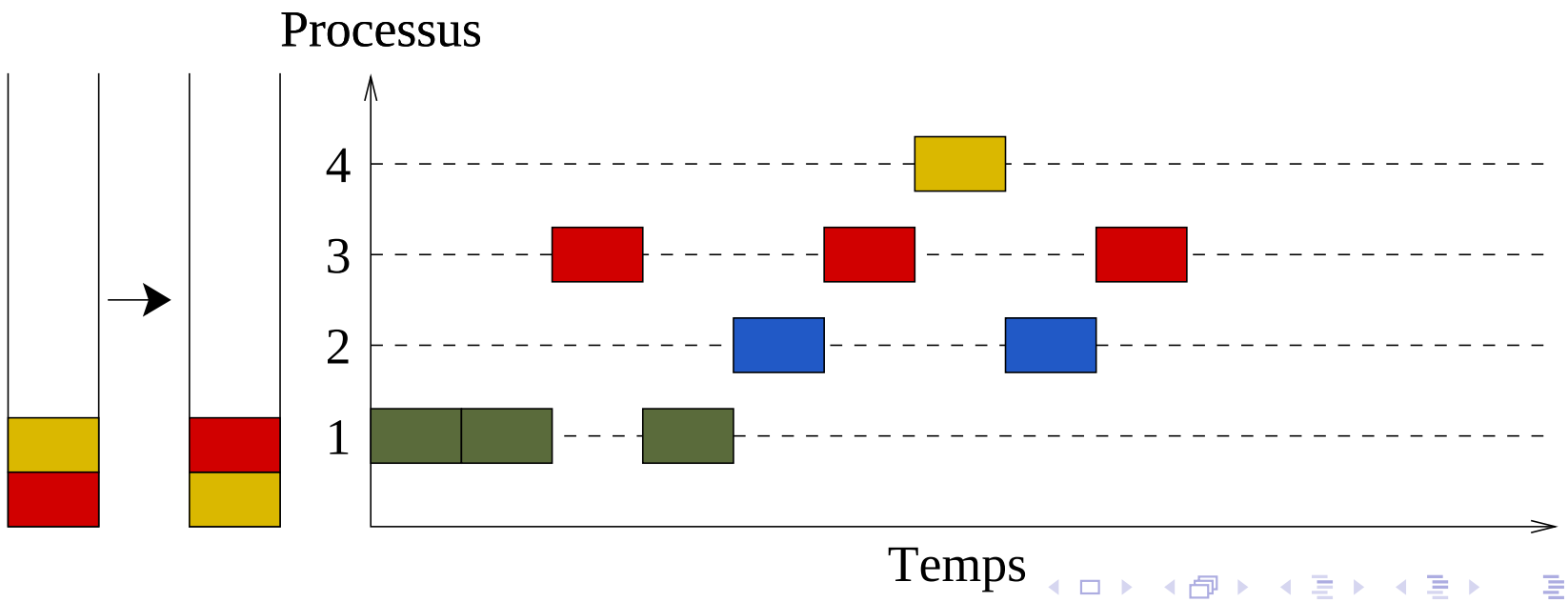


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

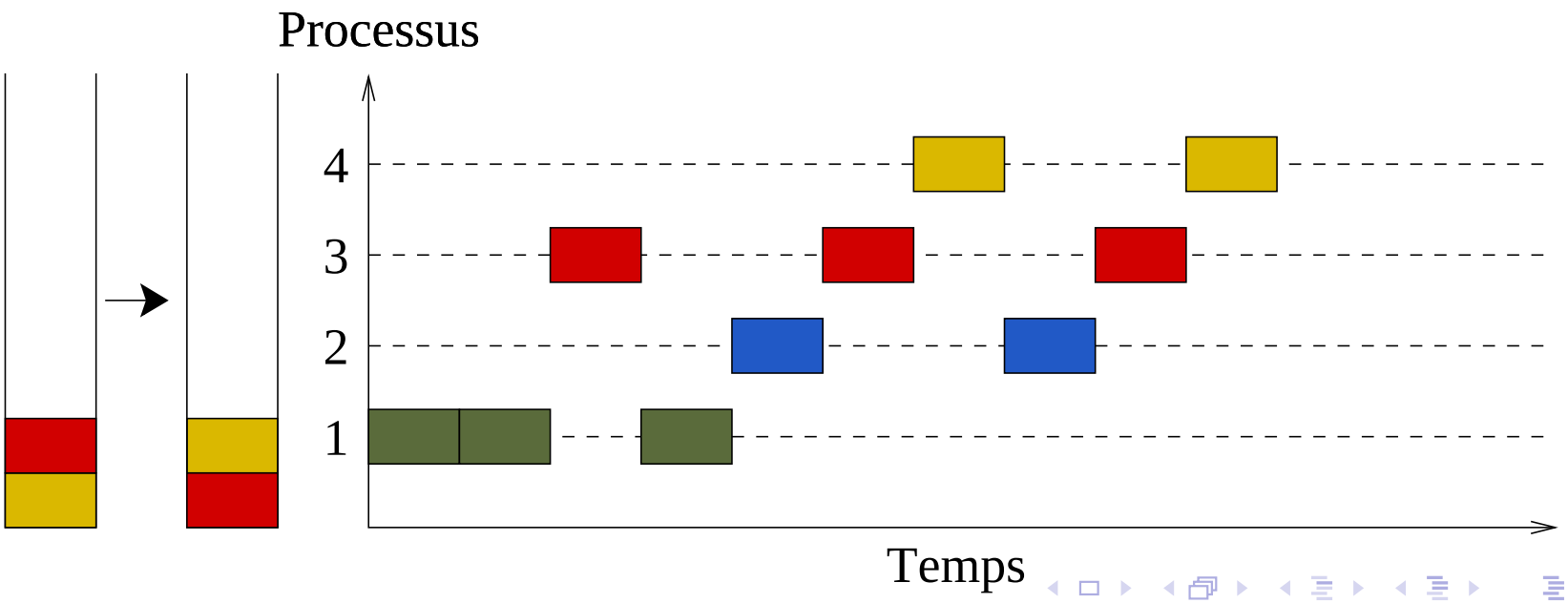


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

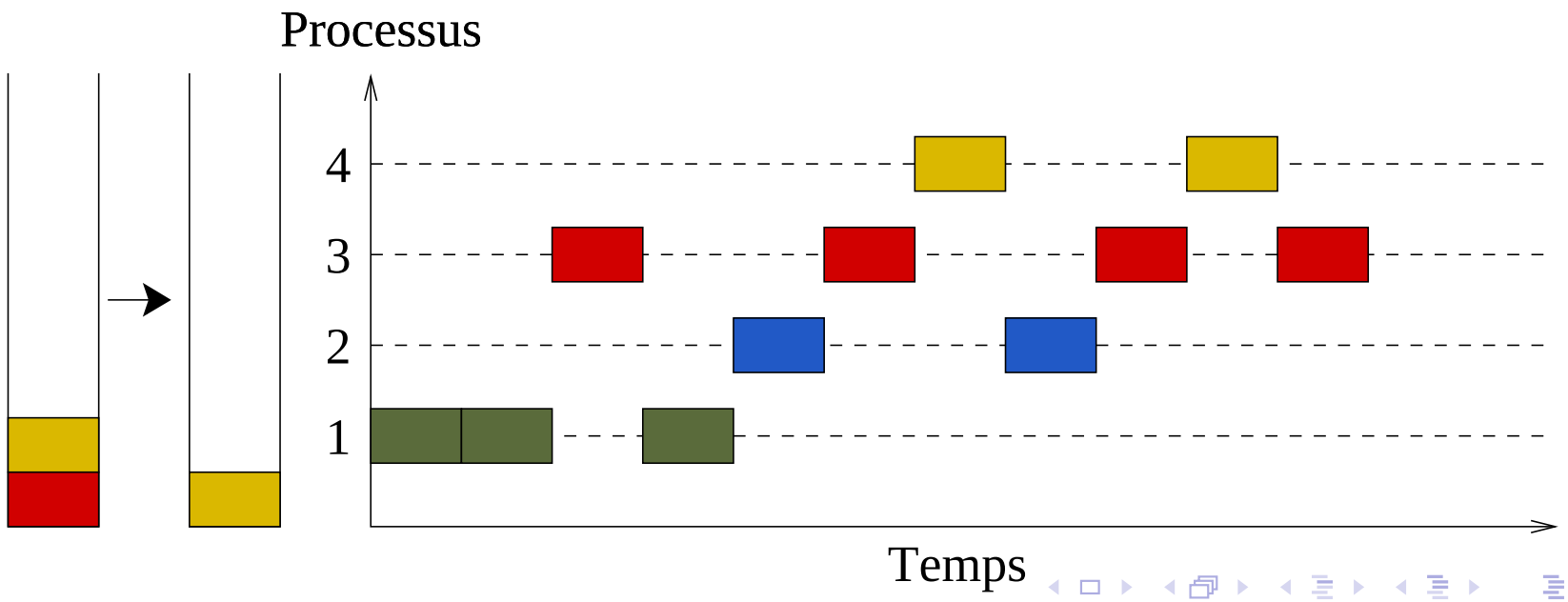


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

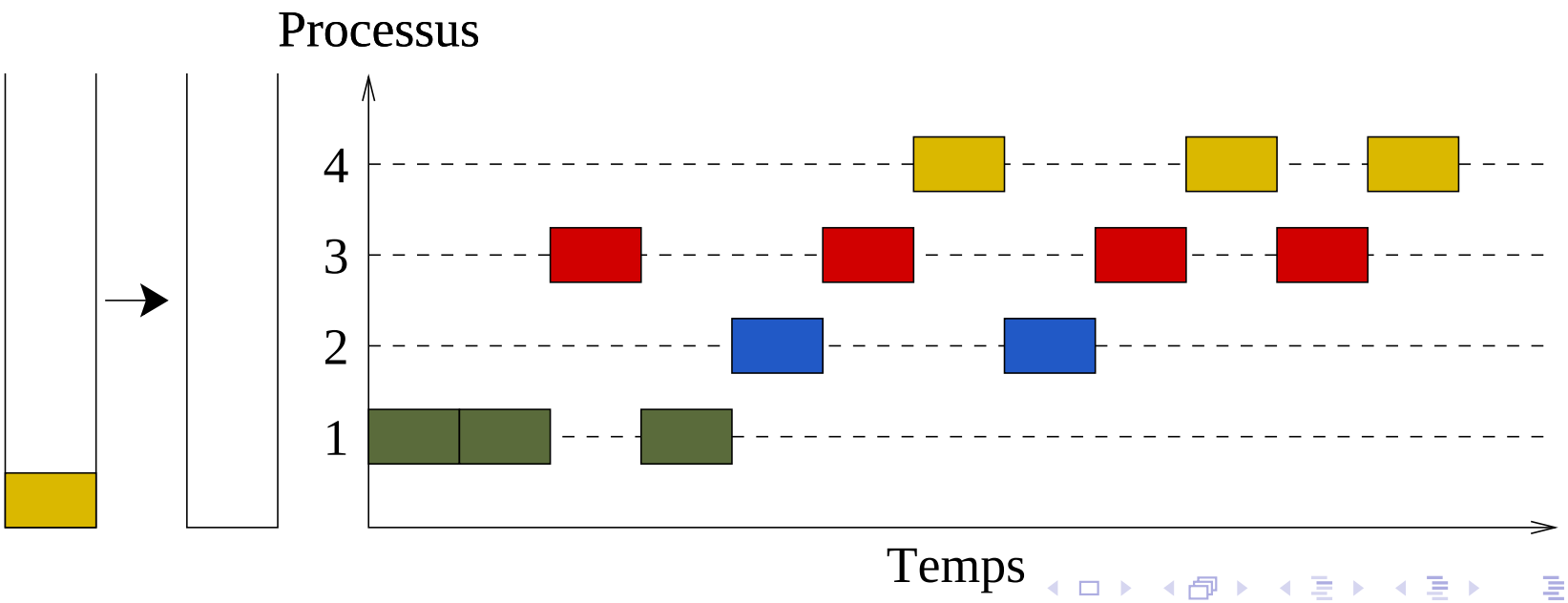


# Round Robin : exemple

## Données des processus

| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1

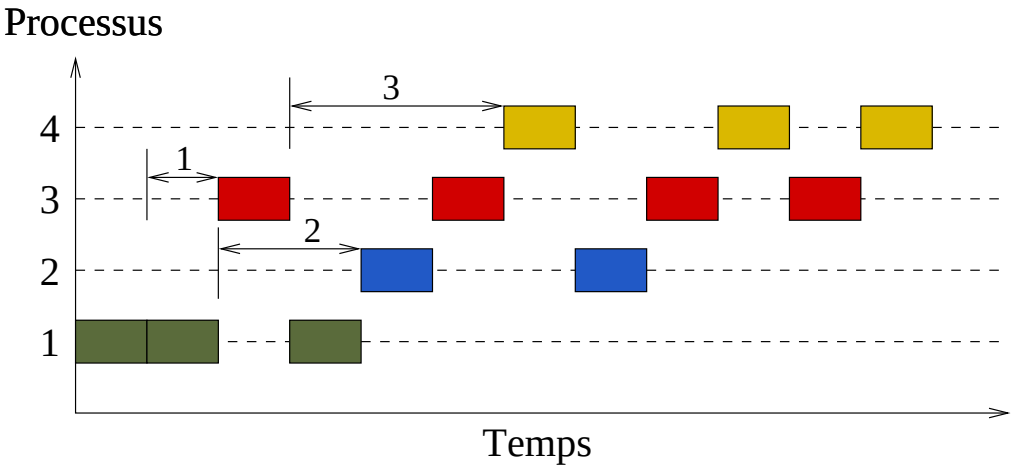


# Round Robin : exemple

## Données des processus

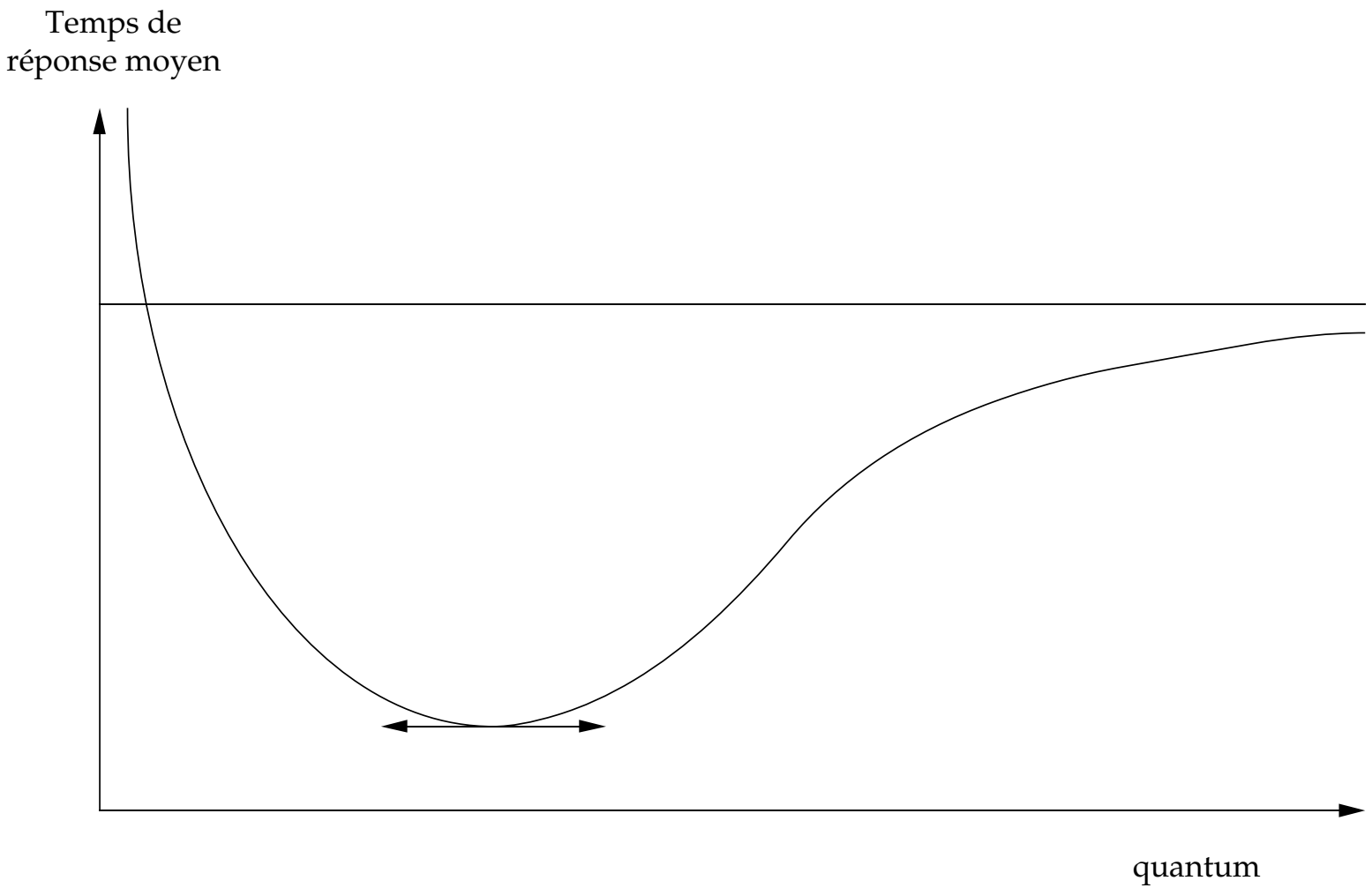
| Processus | Durée | Temps de soumission |
|---|---|---|
| 1 | 3 | 0 |
| 2 | 2 | 2 |
| 3 | 4 | 1 |
| 4 | 3 | 3 |

Quantum = 1




| Processus | Temps d'attente | Temps d'exécution |
|---|---|---|
| 1 | 0 | 4 |
| 2 | 2 | 6 |
| 3 | 1 | 10 |
| 4 | 3 | 9 |
| Moyenne | 1.5 | 7.25 |

# Temps de réponse : Quantum



# Files de priorité

## Description de l'algorithme *Files de priorité*

Il existe une file par ordre de priorité. La gestion de chaque file est faite par l'algorithme Round Robin. Les files sont traitées dans l'ordre décroissant.

## Propriétés de l'algorithme

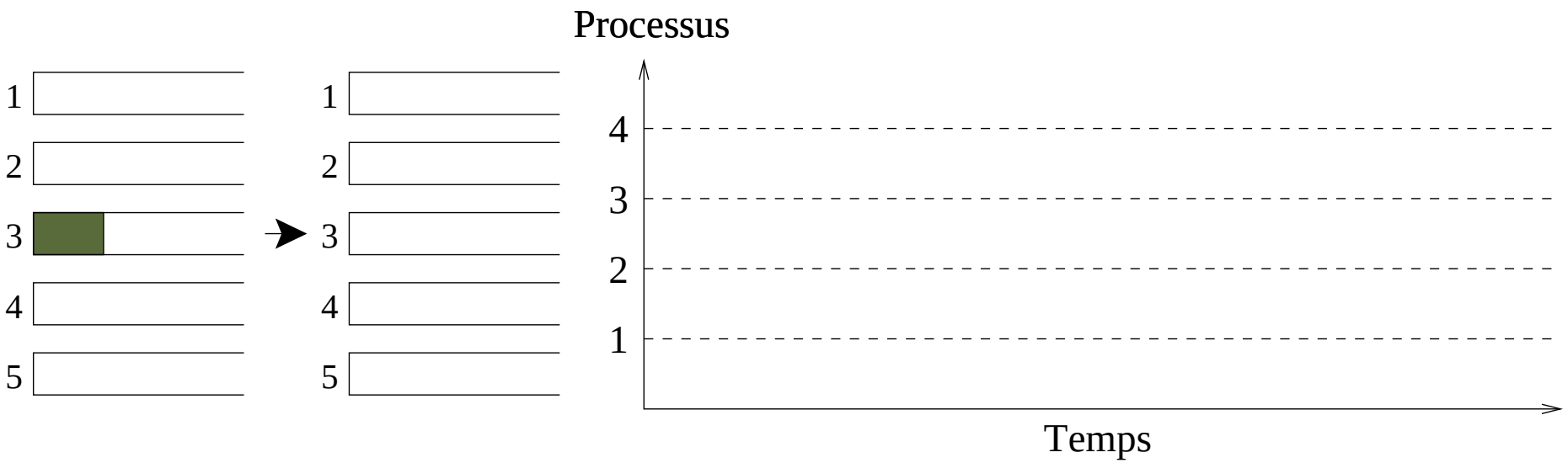
- Notion de priorité sur les processus ;
- Préemptif ;
- Pas de connaissance sur la durée des processus.

# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1

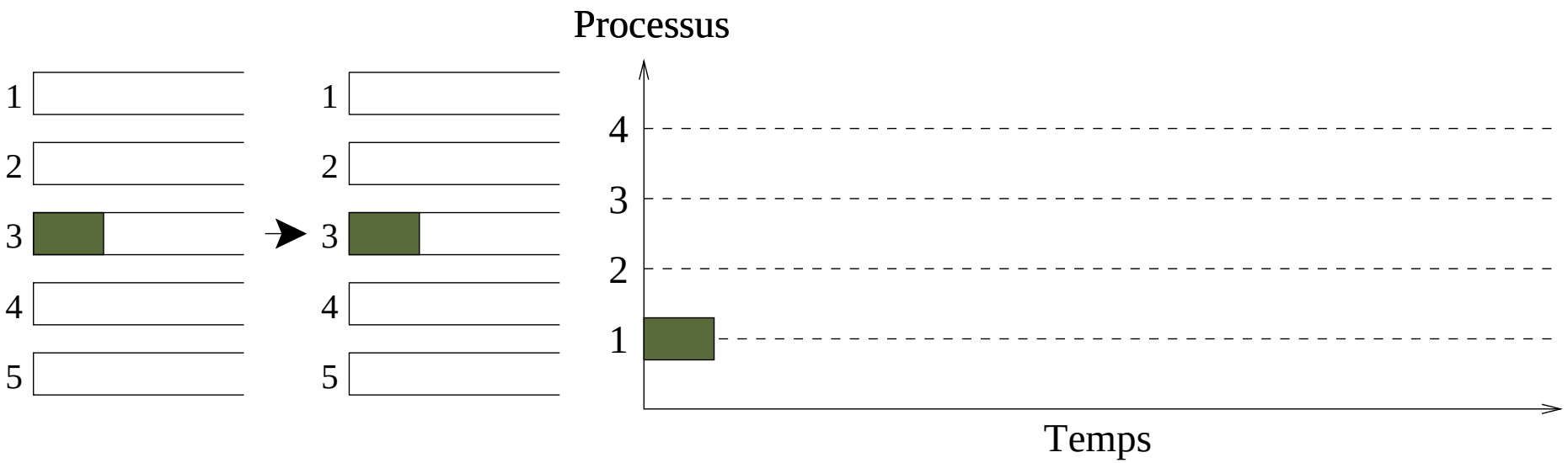


# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1

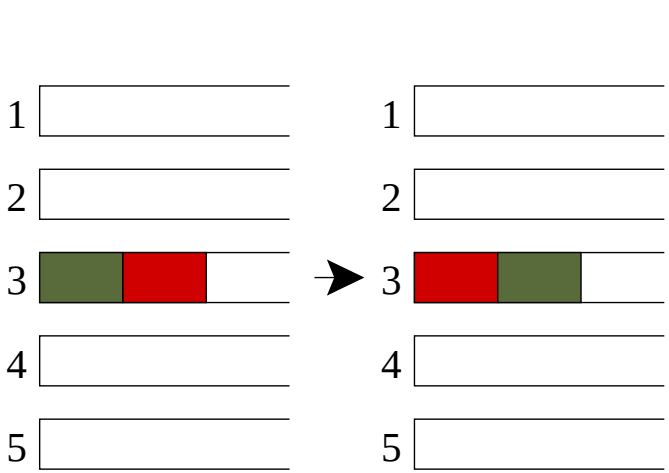


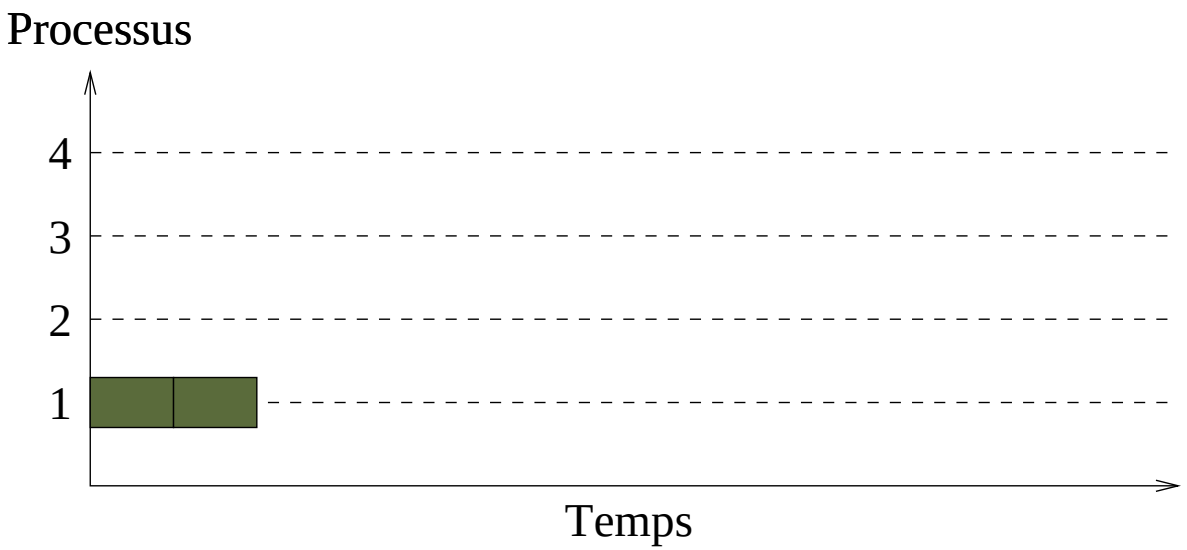
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



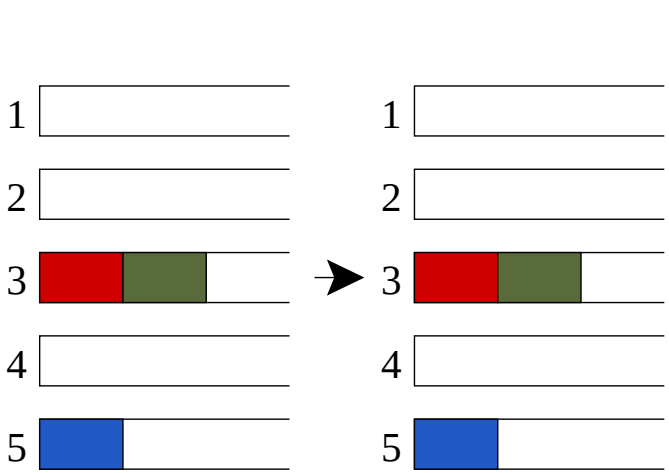



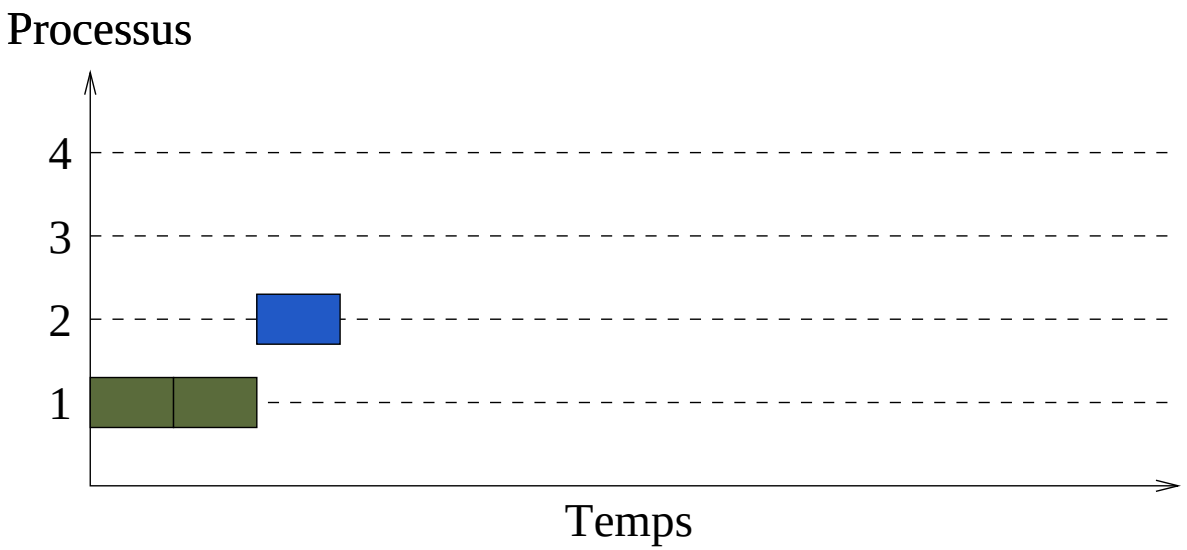
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



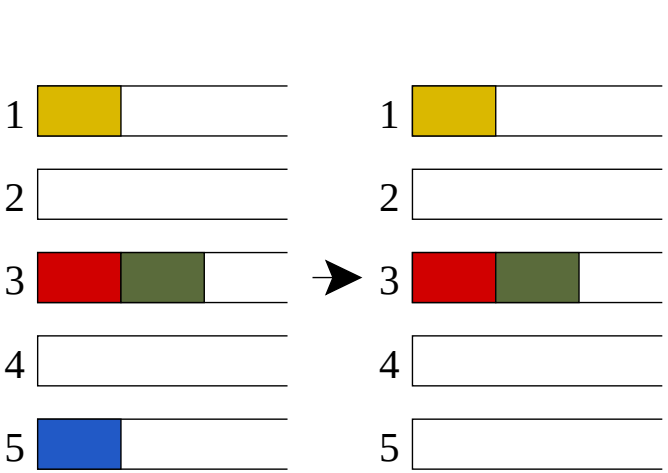



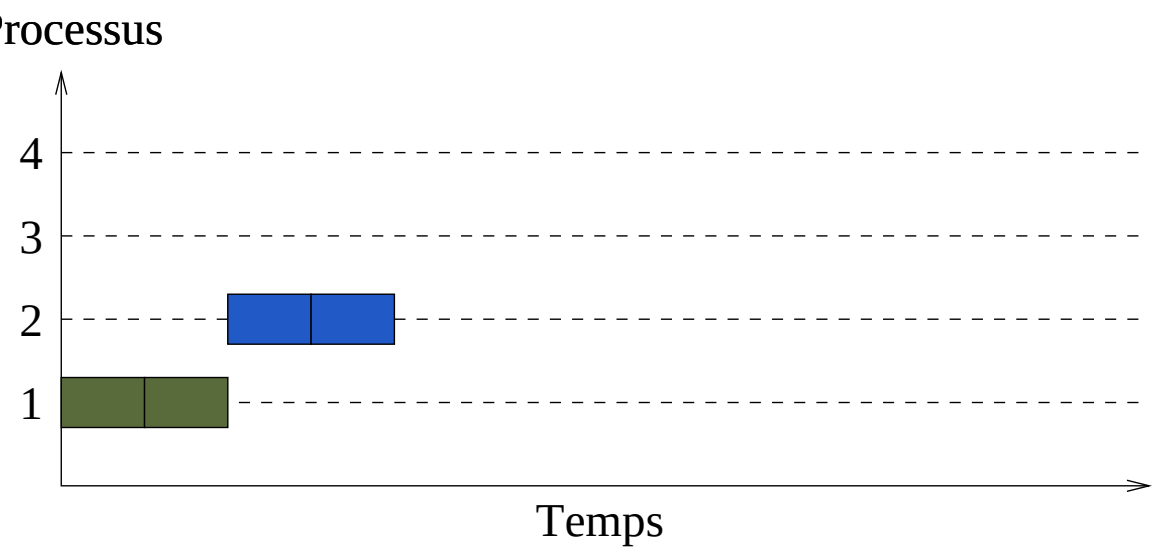
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



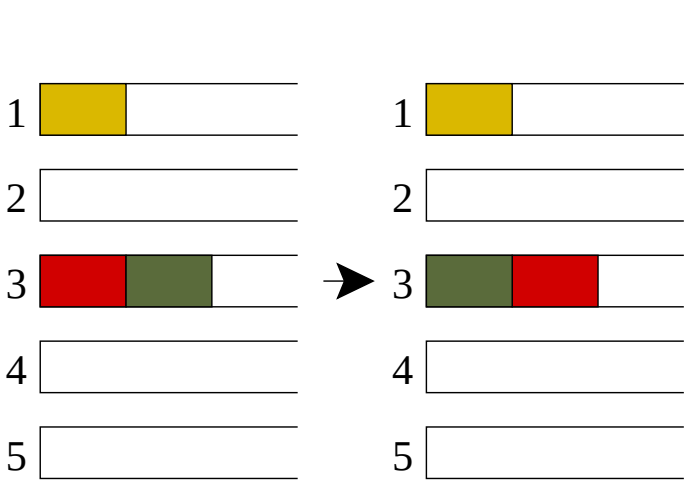



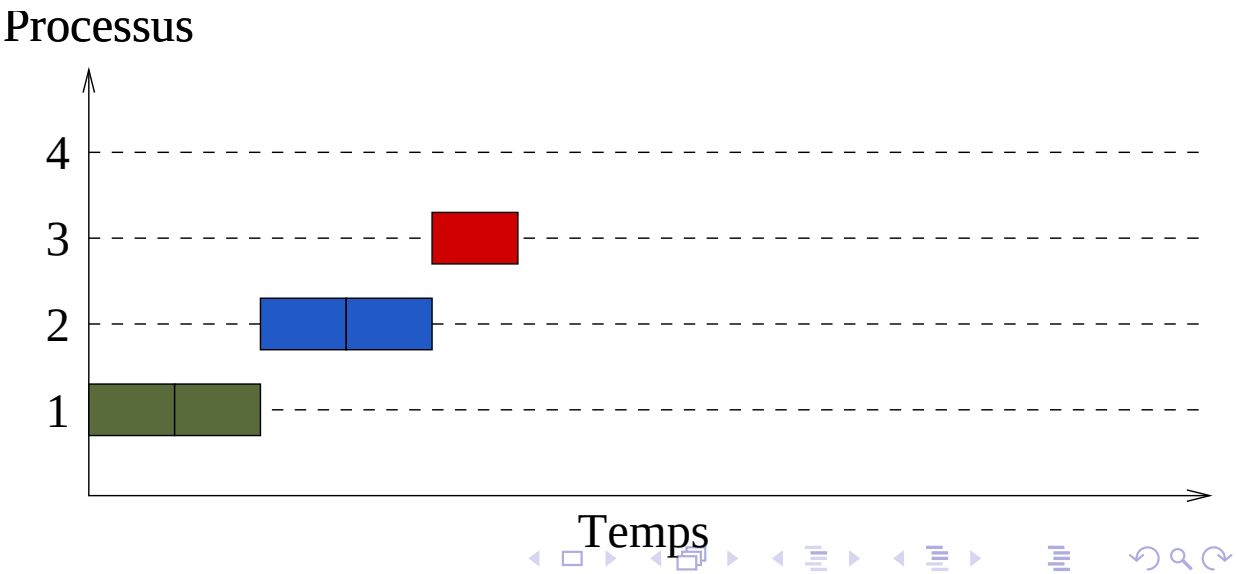
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



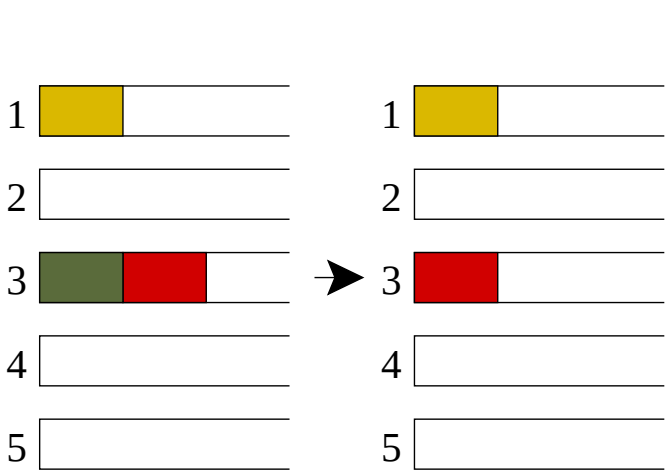



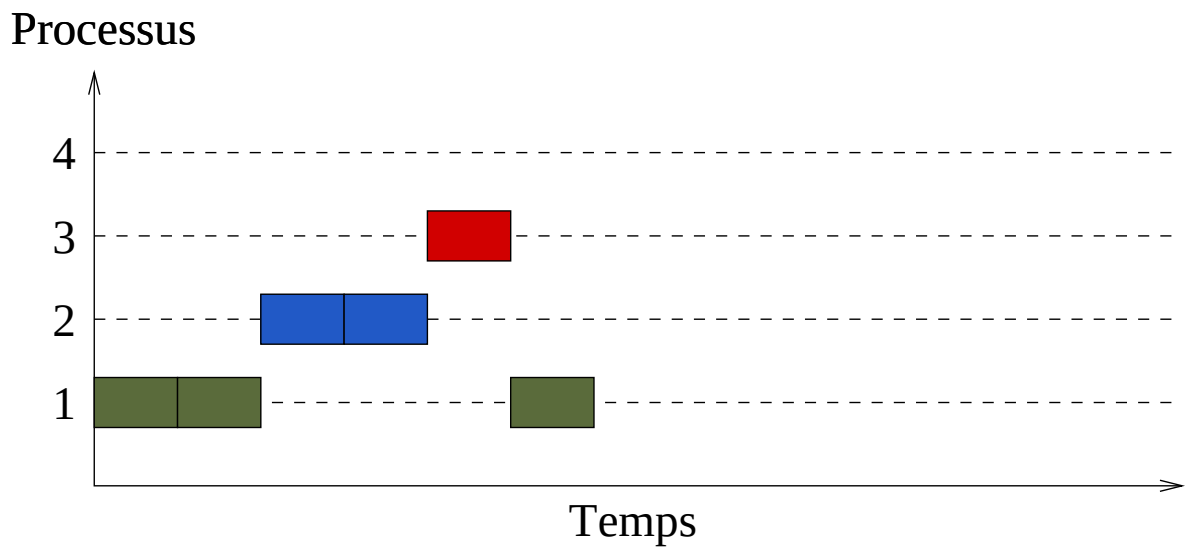
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



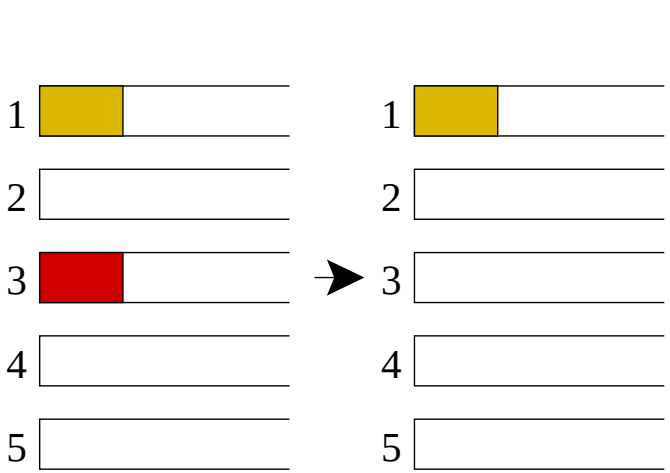



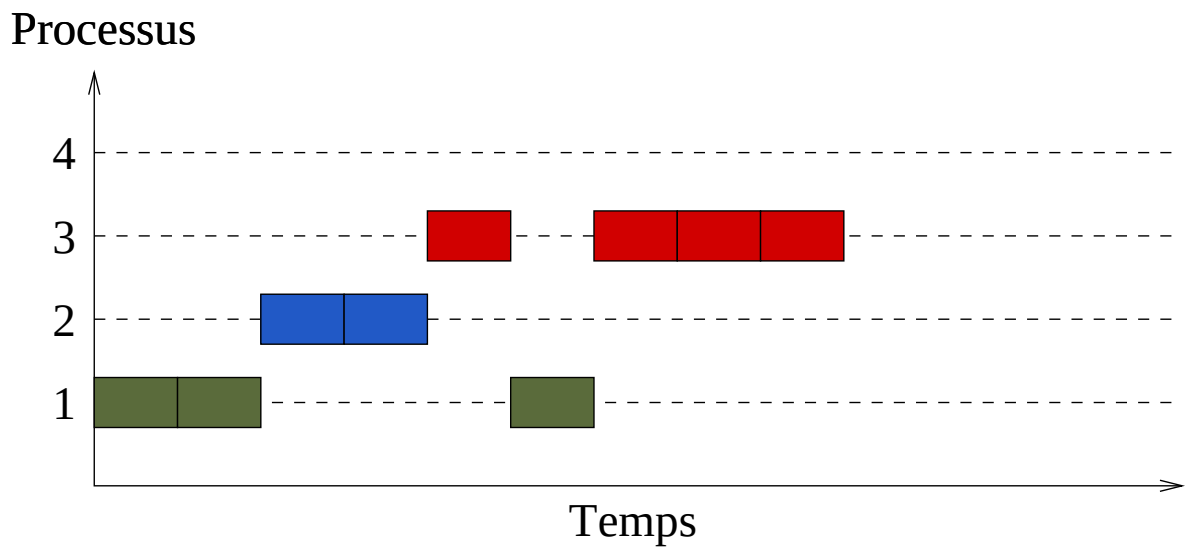
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



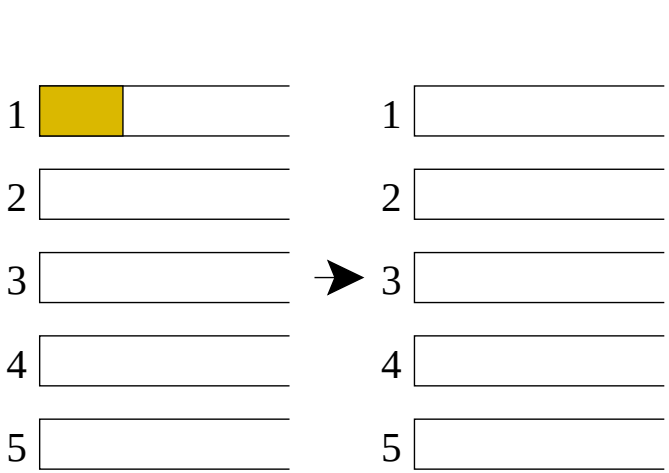



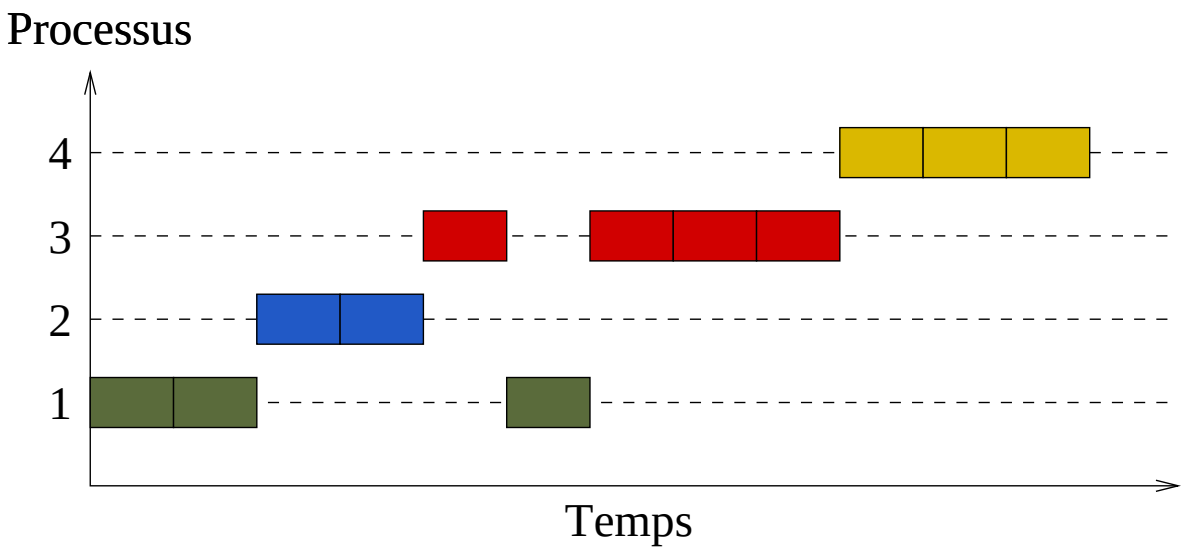
# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1



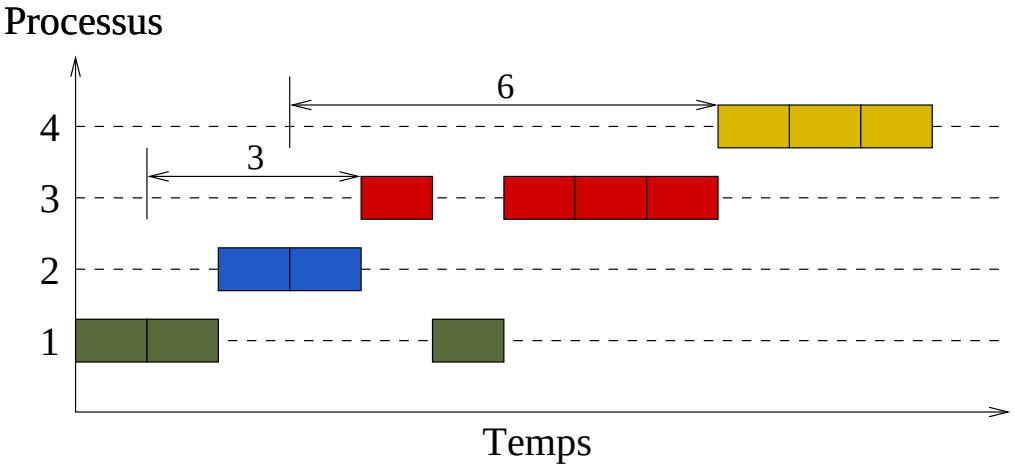



# Files de priorité

## Données des processus

| Processus | Durée | Temps de soumission | Priorité |
|---|---|---|---|
| 1 | 3 | 0 | 3 |
| 2 | 2 | 2 | 5 |
| 3 | 4 | 1 | 3 |
| 4 | 3 | 3 | 1 |

Quantum = 1




| Processus | Temps d'attente | Temps d'exécution |
|---|---|---|
| 1 | 0 | 6 |
| 2 | 0 | 2 |
| 3 | 3 | 8 |
| 4 | 6 | 9 |
| Moyenne | 2.25 | 6.25 |

# Files de priorité dynamique

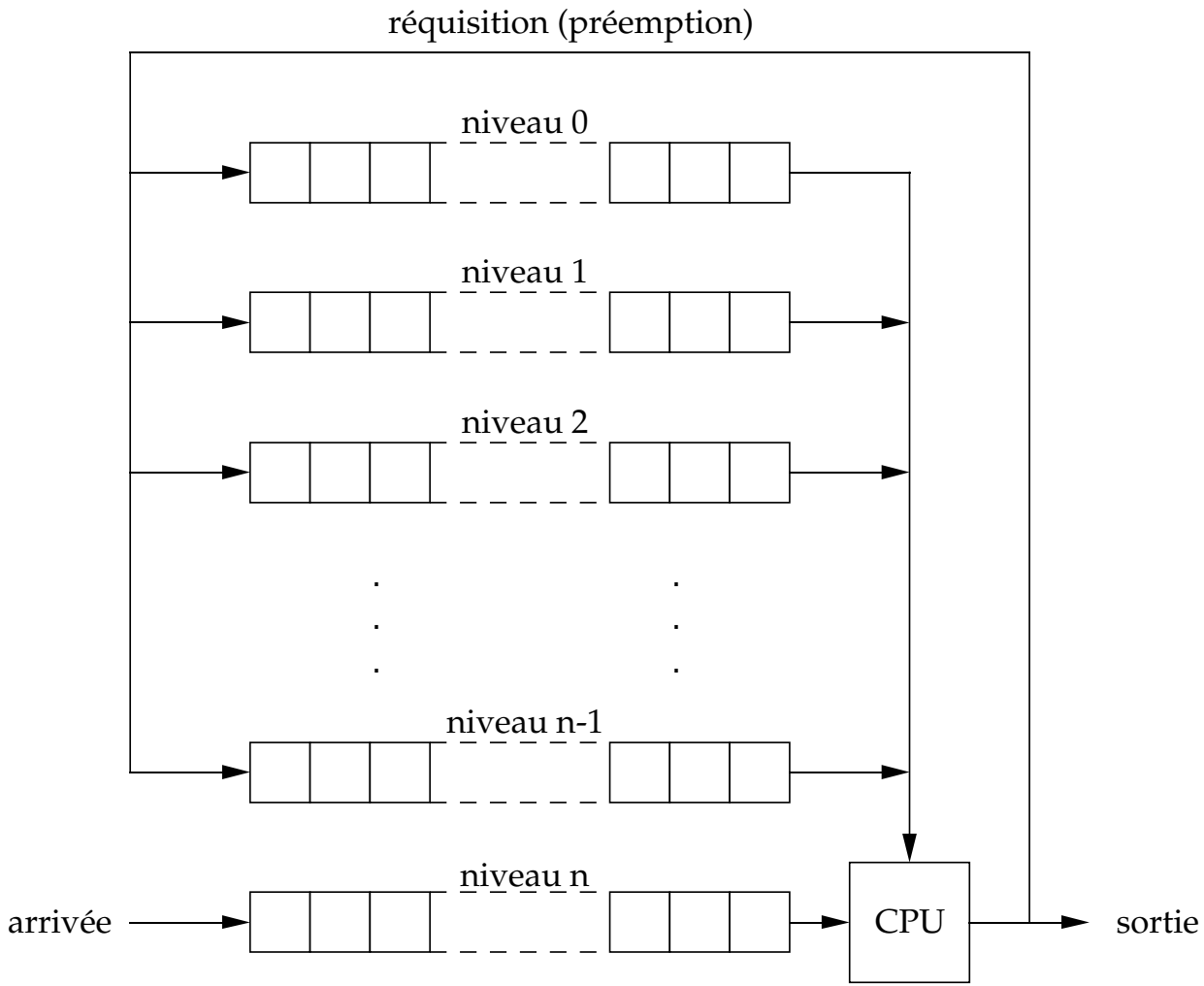
## Description de l'algorithme *Files de priorité dynamique*

Il existe une file par ordre de priorité plus une file pour le nouveau processus. Tout les nouveaux processus sont placés sur ce file plus prioritaire. Après la première exécution, le processus est placé dans une autre file de priorité.

## Propriétés de l'algorithme

- Notion de priorité sur les processus ;
- La priorité est calculé dynamiquement ;
- Différentes équations peuvent être utilisées ;
- Préemptif ;
- Pas de connaissance sur la durée des processus.

# Files de priorité dynamique



# Files de priorité - classes de processus

## Classes de processus

Il existe des classes de processus qui sont rangés dans différent files de priorité.

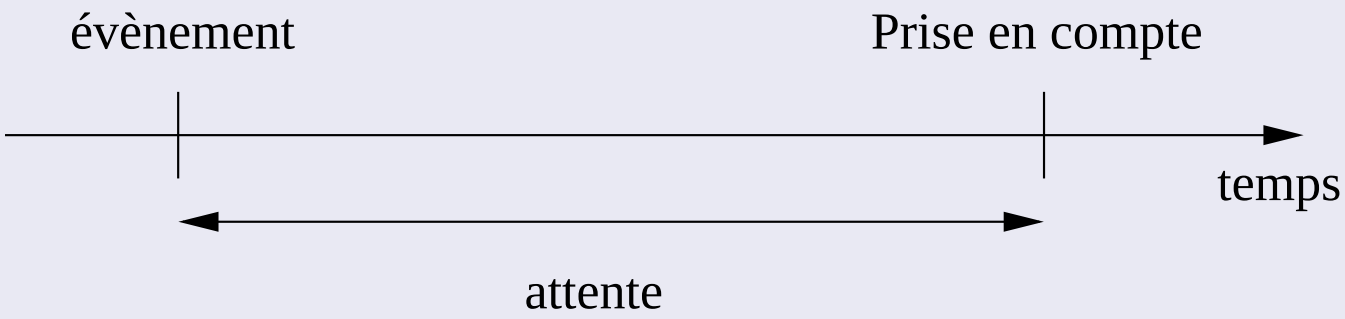
- processus systèmes,
- processus temps réel,
- processus interactifs,
- processus d'édition interactive,
- processus « batch ».

Chaque classe est **absolument** prioritaire sur celles du niveau inférieur.
À l'intérieur de chaque classe, les processus sont rangés dans un système de files de priorité multi-niveaux.

# Processus temps réel

## Le cas des processus temps réel

Prise en compte d'un événement :




si

- $n$ est le nombre de processus temps réel,
- $s$ le temps pris par l'ordonnanceur,
- $t$ le temps moyen pris par les processus T.R.

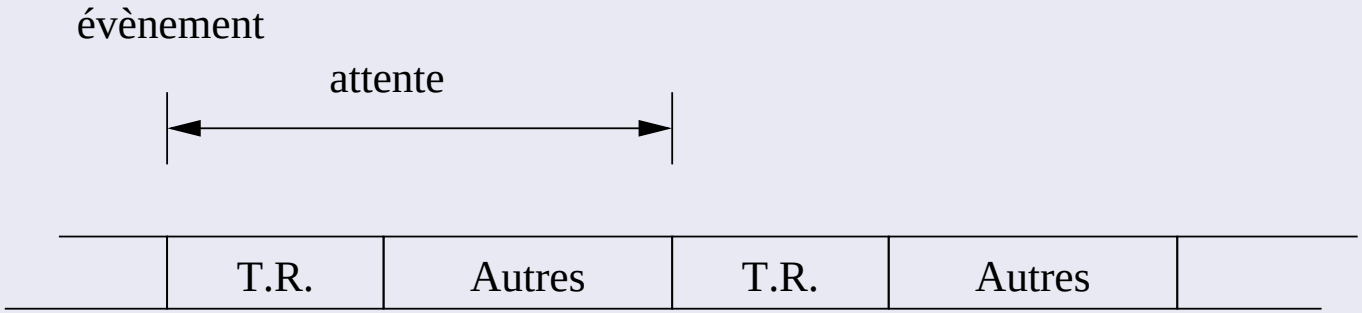
alors

$$(n-1) \times (s+t) < attente$$

# Mélange temps réel et temps machine

## Division du temps CPU

Le temps CPU peut être divisé en une partie fixe réservée au processus Temps Réel, et une partie réservée aux autres processus.




Les algorithmes précédents s'appliquent dans la partie T.R.,
Les algorithmes classiques sont utitilés par la partie « Autres ».

# Allocation multi-processus

## Hypothèse

On dispose de $N$ processeurs identiques.

## Objectifs

- équilibrage de la charge (homogénéité des taux d'utilisation de la CPU,
- évitez les effets de « pompage ».

## Systèmes simples X sophistiqués

- **Système simples** : le S.E. s'exécute toujours sur la même CPU,
- **Système plus sophistiqué** : le S.E. est capable de s'exécuter en parallèle sur plusieurs processeurs.

$\rightarrow$ Il faut gérer les problèmes de synchronisation.

# Table de matière

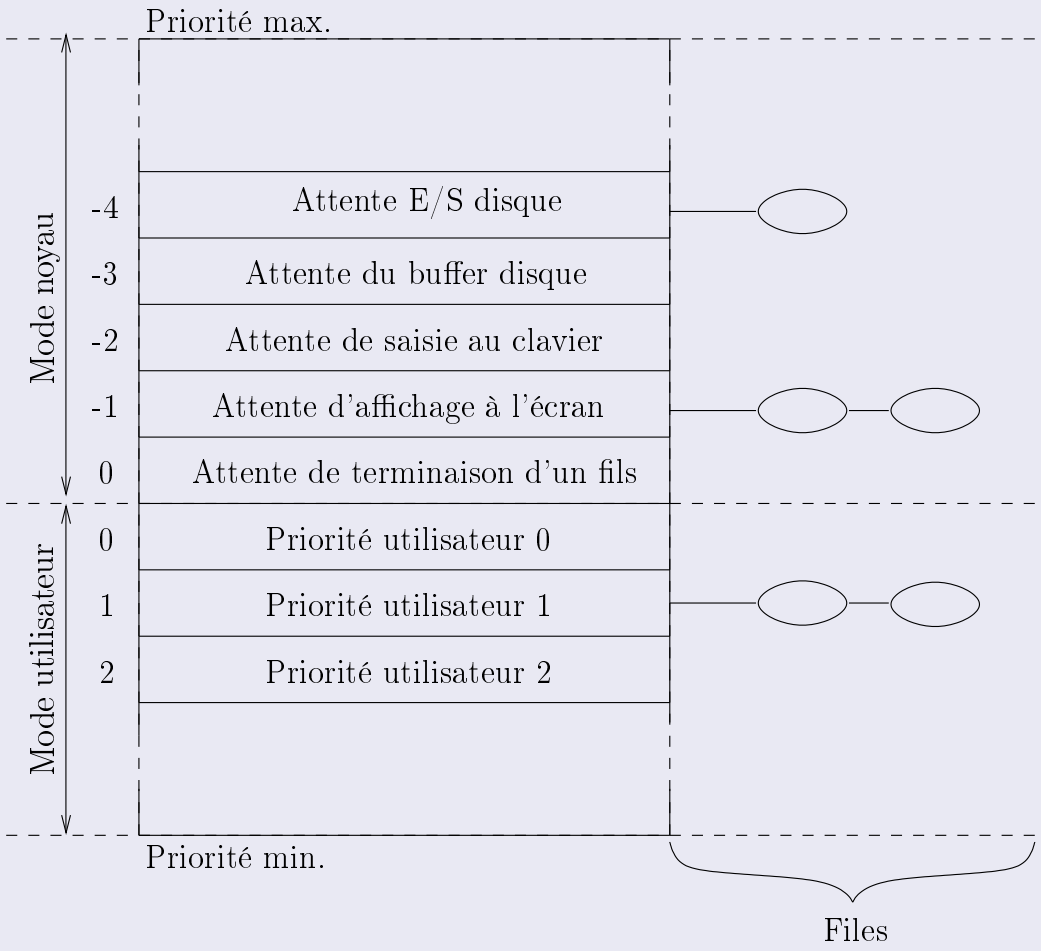
# Généralités

## Ordonnancement sous Unix

- Utilisation de files à plusieurs niveaux, chaque file correspondant à un ensemble de priorités disjoint des autres
- Les processus sont séparés en deux parties :
  - Les processus qui s'exécutent en mode noyau (i.e. processus qui font des appels système)
  - Les processus utilisateurs (i.e. qui ne font pas d'appel système)
- Les processus utilisateurs ont des priorités à valeurs positives
- Les processus qui s'exécutent en mode noyau ont des priorités à valeurs négatives
- Les priorités négatives sont prioritaires par rapport aux valeurs positives
- Uniquement les processus qui résident en mémoire et prêts à l'exécution sont placés dans ces files

# Représentation

## Représentation de l'ordonnanceur Unix



# Algorithme

## Description

- L'ordonnanceur parcourt les files en partant de la priorité la plus haute (valeur la plus basse) ;
- Dès qu'une file non vide est trouvée : le premier processus de la file est démarré ;
- Le processus s'exécute pendant un quantum de temps (typiquement 100 ms) sauf s'il se bloque avant ;
- Une fois le quantum terminé, le processus est placé en fin de file ;
- Les processus de même classe partagent donc le CPU suivant l'algorithme *Round Robin*.

# Mise à jour des priorités

## Description

- Chaque seconde, la priorité de chaque processus est recalculée suivant la formule :

$$\texttt{priorité} = \texttt{usage CPU} + \texttt{nice} + \texttt{base}$$

- `usage CPU` : utilisation moyenne du CPU depuis les dernières secondes
- `nice` : valeur *sympa*, par défaut à 0 mais pouvant varier de -20 à 19 :
  ↪ Processus utilisateur entre 0 et 19
  ↪ Valeurs négatives uniquement pour les processus root
- `base` : priorité de base
- Les processus sont alors replacés dans les bonnes files de priorité

# Généralités (1/2)

## Ordonnancement sous Linux

- L'ordonnancement est différent des systèmes Unix et est basé sur les threads (et non les processus) ;
- Trois classes de threads sont définies pour l'ordonnancement :
  - FIFO temps réel ;
  - Round Robin temps réel ;
  - Threads en temps partagé.

# Généralités (2/2)

## Threads FIFO temps réel

- Non préemtibles ;
- Prioritaires par rapport aux autres (priorités entre -100 et -1).

## Threads Round Robin temps réel

- Idem que la première catégorie excepté que les threads sont interruptibles par l'horloge
- S'il y a plusieurs processus éligibles, ils s'exécutent selon l'algorithme Round Robin

## Thread en temps partagé

- Destinés aux processus non prioritaires.

# Priorité et quantum

## Priorité

- Priorité d'ordonnancement entre 0 et 39 ;
- Peut être modifiée par l'appel système `nice(val)`
- Plus la priorité est grande, plus le processus est rapide à répondre et reçoit une proportion de CPU plus importante

## Quantum

- A chaque thread est associé un quantum de temps (nombre de tops d'horloge)
- L'ordonnanceur calcule une valeur de *bonté* d'un thread :
  - Si (classe = temps réel) alors bonté $\leftarrow$ 1000 + priorité ;
  - Si (classe = temps partagé et quantum $>$ 0) alors bonté $\leftarrow$ quantum + priorité ;
  - Si (classe = temps partagé et quantum = 0) alors bonté $\leftarrow$ 0

# L'ordonnancement

## Sélection d'un processus

- Le thread possédant la plus forte valeur bonté est sélectionné ;
- Pendant son exécution, son quantum est décrémenté de 1 ;
- Un thread est alors retiré du CPU si :
  - Son quantum est à 0 ;
  - Il est bloqué par un appel d'entrée/sortie, un sémaphore ou autre ;
  - Un thread de bonté supérieure jusque-là bloqué devient éligible.

## Mise à jour des quanta

- Au fur et à mesure de l'exécution des threads, leur quantum va donc diminuer jusqu'à 0 ;
- Les threads bloqués, quand à eux, auront un quantum différent de 0 ;
- Une fois qu'aucun thread n'est plus éligible, on fait le mise à jour des quanta de tous les processus :

```
quantum = (quantum / 2) + priorité
```

# Conséquences de l'algorithme

## Description

- Un thread utilisant beaucoup le CPU :
  - Son quantum diminue rapidement ;
  - La nouvelle valeur sera égale à la priorité
- Un thread effectuant beaucoup d'entrées/sorties :
  - Son quantum ne sera pas égal à 0
  - La valeur de son quantum va augmenter (jusqu'à sa priorité $\times 2$)
- Si plusieurs threads utilisant de la CPU sont en concurrence :
  - Leur quantum diminue vers priorité
  - Les processus les plus prioritaires obtiennent une portion plus importante du CPU

# Principe général

## Description

- Pas de thread central pour l'ordonnancement ;
- Lorsqu'un thread ne peut plus s'exécuter . . .
  - Bloqué par une sémaphore, un mutex, une entrée/sortie ;
  - Signale un objet (libération d'une sémaphore) ;
  - Le quantum de temps est expiré.
- . . . il entre en mode noyau et lance lui-même l'ordonnanceur :
  - Il sauvegarde son propre contexte ;
  - Lance l'ordonnanceur pour trouver le successeur (et le charger).

# Niveaux de priorité

## Priorité

- La priorité du thread au sein de son processus :
  - time critical (temps critique), highest (le plus haut), above normal (au-dessus de la normale), normal, below normal (en-dessous de la normale), lowest (le plus bas), idle (inactif)
- La priorité des processus :
  - realtime (temps réel), high (haut), above normal, normal, below normal, idle

## Répartition des priorités sur 32 niveaux

| | realtime | high | above normal | normal | below normal | idle |
|---|---|---|---|---|---|---|
| time critical | 31 | 15 | 15 | 15 | 15 | 15 |
| highest | 26 | 15 | 12 | 10 | 8 | 6 |
| above normal | 25 | 14 | 11 | 9 | 7 | 5 |
| normal | 24 | 13 | 10 | 8 | 6 | 4 |
| below normal | 23 | 12 | 9 | 7 | 5 | 3 |
| lowest | 22 | 11 | 8 | 6 | 4 | 2 |
| idle | 16 | 1 | 1 | 1 | 1 | 1 |

# Gestion des priorités

## Description

- Utilisation d'un tableau à 32 entrées (1 par priorité)
- Chaque entrée pointe vers une liste de threads en attente avec cette priorités
- L'algorithme recherche la première entrée non vide
- Le thread en tête est exécuté pour un quantum de temps
- A la fin du quantum, le thread est placé en fin de file

# Évolution des priorités

## Description

- 4 niveaux réels de priorités : temps réel (abus de langage !), utilisateur, zéro et idle.
- Les priorités temps réel sont réservées pour le système ou aux threads autorisés par l'administrateur
- Le thread zéro s'exécute en tâche de fond et utilise le temps CPU restant pour le gestionnaire de mémoire
- La priorité d'un thread utilisateur peut augmenter :
  - A la fin d'une opération d'entrée/sortie (augmentation suivant le périphérique)
  - En attente d'une sémaphore qui s'est libérée (priorité augmentée de 2)
- Cette augmentation est temporaire : elle prend effet immédiatement mais si un thread a épuisé son quantum de temps, l'augmentation est réduite de 1 point (et ainsi de suite jusqu'au niveau de priorité du thread)