

# **TP - Système d' Exploitation Avancé**

**Chef de projet :** Chongguang LIU

Elody CATINEL  
Yannick FAULA  
Iler VIRARAGAVANE  
Florent BOISSELIER  
Franck MPEMBA BONI

H4103

# Ordonnancements

- Collaboratif
- Préemptif
- Préemptif à priorités

# Ordonnanceurs

Analyse théorique des algorithmes

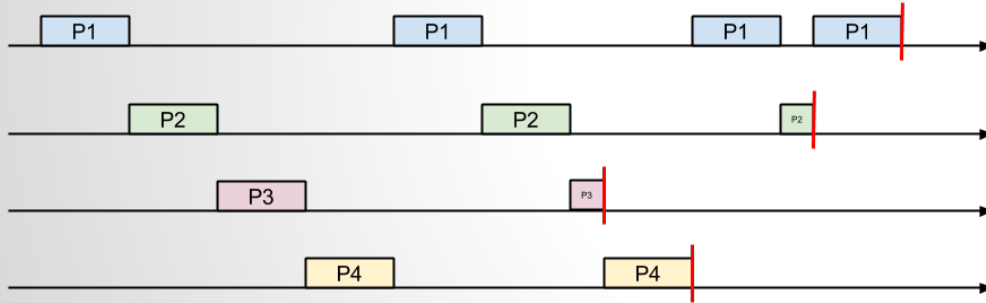
## Exemple de test

	Priorité	Durée	Quantum
P1	10	12	10
P2	5	7	5
P3	2	4	3
P4	2	6	3

# Ordonnanceurs

## Analyse théorique des algorithmes

- Préemptif sans priorité

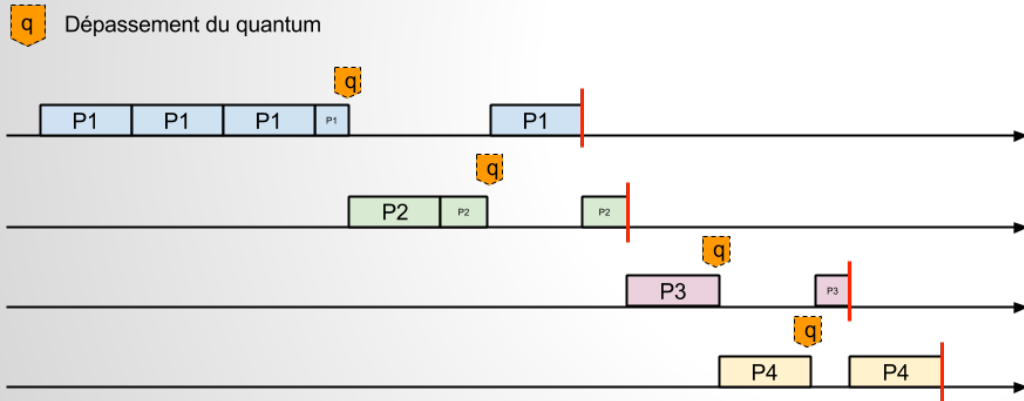


	Temps de réponse	Temps d'attente	Temps d'exécution
Moyenne	4,5	17,25	24,5

# Ordonnanceurs

## Analyse théorique des algorithmes

- Préemptif à priorité standard




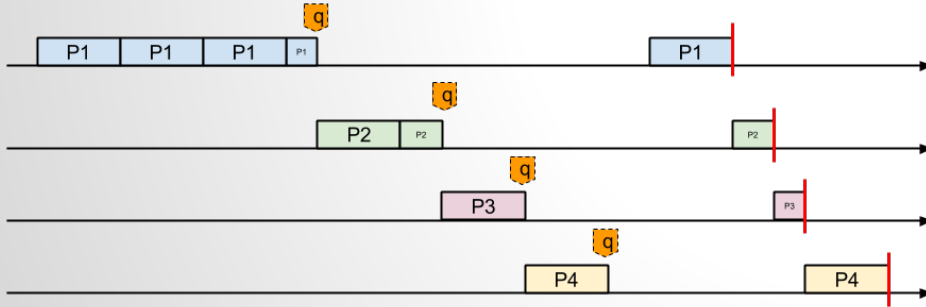
	Temps de réponse	Temps d'attente	Temps d'exécution
Moyenne	12,75	14,75	22

# Ordonnanceurs

## Analyse théorique des algorithmes

- Préemptif à priorité “famine réduite”

 Dépassement du quantum



	Temps de réponse	Temps d'attente	Temps d'exécution
Moyenne	10,75	18,5	25,75

# Ordonnanceurs

## Analyse théorique des algorithmes

### Préemptif sans priorité

Bon temps de réponse, équilibre

	Temps de réponse	Temps d'attente	Temps d'exécution
Moyenne	4,5	17,25	24,5

### Préemptif à priorités standard

Priorités et terminaison au plus vite

	Temps de réponse	Temps d'attente	Temps d'exécution
Moyenne	12,75	14,75	22

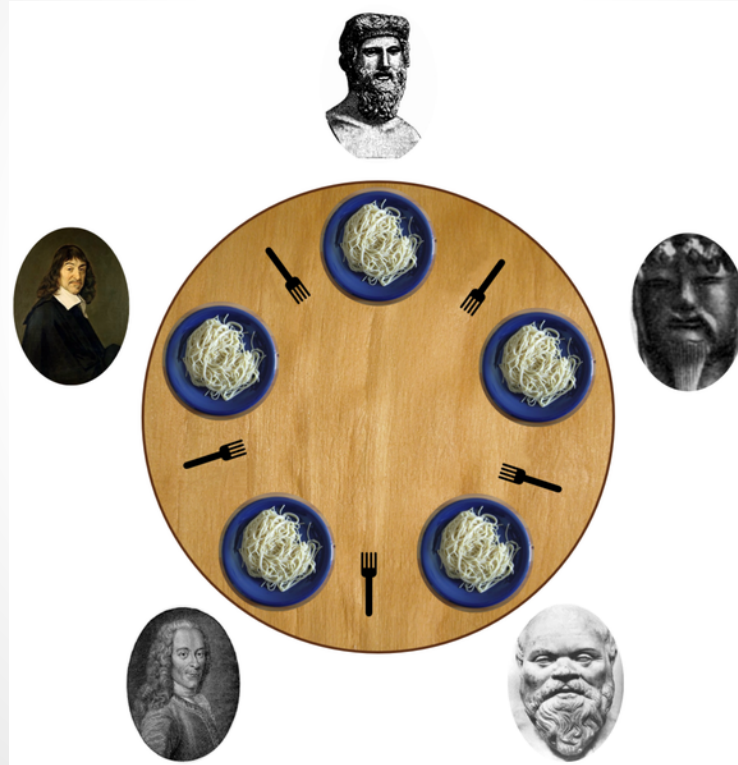
### Préemptif à priorités “famine réduite”

Priorités et évolution en parallèle de tous les processus

	Temps de réponse	Temps d'attente	Temps d'exécution
Moyenne	10,75	18,5	25,75

# Problème des philosophes

Démo :





# Problème des philosophes

- **Détection de interblocages**
  - Philosophes seuls
  - Avec un autre processus
- **Prévention des interblocages**
  - Libération automatique des fourchettes
  - Blocage des interruptions

# Lecteur MIDI

Démo :

- Avec ordonnanceur Préemptif
  - Problème de **Waiting Time**
- Avec ordonnanceur Préemptif avec priorité

# Ordonnanceur Linux

Démo :

- Temps partagé sans priorités (OTHER)

priorité dynamique = 0

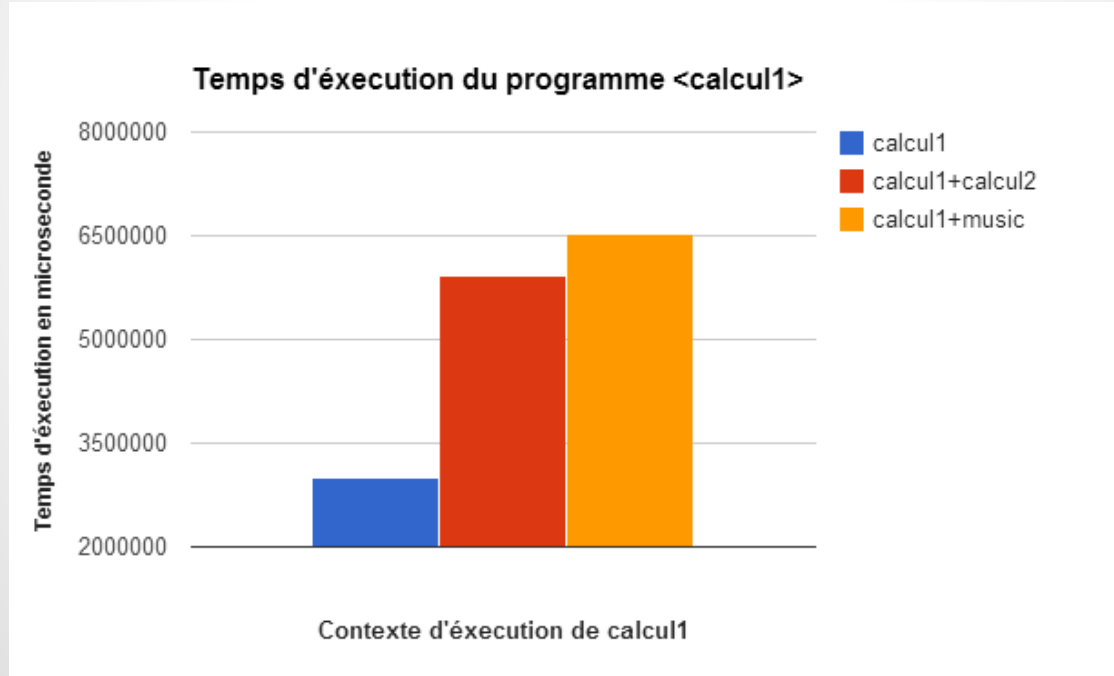
- Temps réel avec priorités (FIFO et RR)

priorité statique variant entre 1 et 99

# Ordonnanceur Linux

- Priorité selon les types de programme

I/O Bound vs. CPU Bound



# Ordonnanceur Linux

- La stratégie de I/O Bound n'est pas parfaite