



Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition

Critères

Algorithmes

FCFS

SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

# Systèmes d'Exploitation

## Ordonnancement des processus

Didier Verna

[didier@lrde.epita.fr](mailto:didier@lrde.epita.fr)

<http://www.lrde.epita.fr/~didier>



# Table des matières

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF  
Priorités  
Tourniquet  
Multi-niveau  
Loterie  
Temps Réel

## 1 Généralités

- Ordonnancement et réquisition
- Critères d'ordonnancement

## 2 Algorithmes d'ordonnancement

- Premier arrivé premier servi
- Plus court d'abord
- Ordonnancement avec priorités
- Tourniquet
- Files d'attentes multi-niveau
- Loterie
- Ordonnancement temps-réel



# Qu'est-ce que l'ordonnancement ?

Commencer dès qu'un processus se bloque

Systèmes  
d'Exploitation

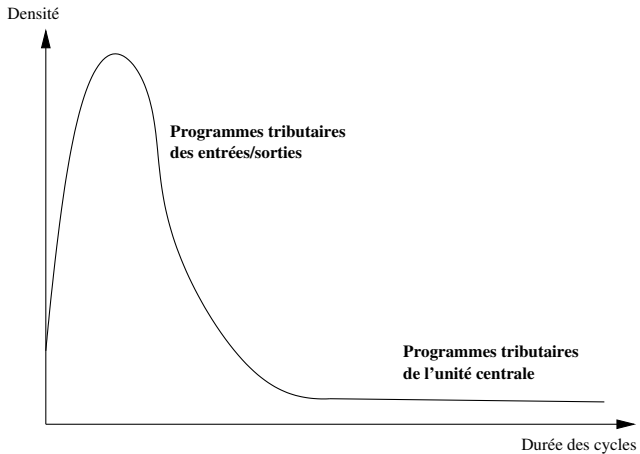
Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF  
Priorités  
Tourniquet  
Multi-niveau  
Loterie  
Temps Réel





# Ordonnancement et Réquisition

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition

Critères

Algorithmes

FCFS

SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Motifs de commutation

- ▶ **Aucun choix** : blocage ou terminaison d'un processus
- ▶ **Choix** : arrivée d'un nouveau processus, passage des états actif ou bloqué à l'état prêt

## ■ Types d'ordonnanceurs

- ▶ **Ordonnancement sans réquisition** (système coopératif) : ne gère que le premier type de commutation. Windows (< 95), Windows NT, Mac OS (< 10).
- ▶ **Ordonnancement avec réquisition** (système préemptif) : gère tous les motifs de commutation. Nécessite des outils de synchronisation et du matériel spécifique (horloge). Windows 95, Mac OS X, UNIX.



# Critères d'ordonnancement

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition

Critères

Algorithmes

FCFS

SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Tous systèmes

- ▶ *Équité* : répartition du CPU
- ▶ *Respect de politique* : imposer les choix d'ordonnancement
- ▶ *Équilibre* : occupation de toutes les parties du système

## ■ Batch

- ▶ *Capacité de traitement / rendement* : nombre de processus exécutés par unité de temps
- ▶ *Temps de restitution / service* : délai entre la soumission d'un processus et sa terminaison (mise en mémoire, attente en état prêt, attente E/S, exécution)
- ▶ *Utilisation du processeur*



# Critères d'ordonnancement (suite)

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition

Critères

Algorithmes

FCFS

SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Interactifs

- ▶ *Temps de réponse* : délai entre la soumission et le moment où l'on commence à répondre
- ▶ *Temps d'attente* : temps passé en état prêt
- ▶ *Proportionnalité* : aux attentes des utilisateurs

## ■ Temps-réel

- ▶ *Respect des dates limites* : éviter la perte de données
- ▶ *Prédictibilité* : stabilité des applications multimédia

⇒ Optimisation min, max, moyenne, variance *etc.*



# Premier arrivé, premier servi (FCFS)

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS

SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Principe

- ▶ Algorithme sans réquisition
- ▶ File d'attente FIFO pour les processus prêts
- ▶ Facile à comprendre et à programmer
- ▶ Intrinsèquement équitable pour des processus équivalents

## ■ Inconvénients

- ▶ Grande variance des critères d'ordonnancement
- ▶ Effet d'accumulation

⇒ Mauvais algorithme pour les systèmes en temps partagé. OK pour les systèmes de batch.



# Plus court d'abord / ensuite (SJF / N)

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Principe

- ▶ Algorithme sans réquisition
- ▶ Le prochain cycle le plus court est sélectionné
- ▶ En cas d'égalité, on revient au FCFS
- ▶ Version avec réquisition : « temps restant le plus court » (SRTF)

## ■ Avantage

- ▶ Temps moyen d'attente minimal

## ■ Inconvénient

- ▶ Difficulté de calculer la longueur des cycles
- ▶ Approximation possible par moyenne exponentielle :  
$$\tau_{n+1} = \alpha \tau_n + (1 - \alpha) \tau_{n-1}$$

⇒ Peu adapté pour l'ordonnancement à court terme. OK pour les systèmes de batch.





# Ordonnancement avec priorités

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités

Tourniquet  
Multi-niveau  
Loterie  
Temps Réel

## ■ Principe

- ▶ Généralisation du SJF (priorité = inverse de la longueur du prochain cycle)
- ▶ Algorithme avec ou sans réquisition
- ▶ Priorités internes (consommation de ressources *etc.*)
- ▶ Priorités externes (fixées par l'utilisateur)

## ■ Inconvénient

- ▶ Blocage infini (« famine »)
- ▶ Solution : technique du « vieillissement » (augmentation progressive de la priorité des processus en attente)



# Tourniquet (RR)

Conçu spécialement pour le temps partagé

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités

Tourniquet

Multi-niveau

Loterie  
Temps Réel

## ■ Principe

- ▶ FCFS avec réquisition sur une base de quanta (20 – 50ms)
- ▶ Nécessite une horloge

## ■ Précautions

- ▶ Le quantum doit être grand par rapport au temps de commutation
- ▶ Le quantum ne doit pas être trop grand

⇒ Réquisition pour les cycles plus longs que le quantum, commutation passive (FCFS) pour les cycles plus courts.



# Files d'attente à plusieurs niveaux

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités  
Tourniquet

Multi-niveau

Loterie  
Temps Réel

## ■ Principe

- ▶ Découpage de la file d'attente des processus prêts en plusieurs files (processus système, interactifs, arrière-plan *etc.*)
- ▶ Ordonnancement spécifique au sein de chaque file (RR, FCFS)
- ▶ Ordonnancement des files entre elles (priorités fixes, allocation de tranches de temps *etc.*)

## ■ Ordonnancement avec feedback (recyclage)

- ▶ Possibilité de déplacer les processus d'une file d'attente à l'autre
- ▶ implémentation du vieillissement
- ▶ dégradation des priorités (ex. cycles longs)



# Ordonnancement par loterie

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Principe

- ▶ Distribution de tickets (CPU, mémoire...)
- ▶ Tirage du gagnant à intervalle fixe

## ■ Avantages

- ▶ Implémentation légère d'un mécanisme de « promesse » (les processus importants peuvent obtenir plusieurs tickets)
- ▶ Efficace pour des processus coopératifs (transmission de tickets)



# Catégories d'événements temps réel

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités

Tourniquet

Multi-niveau

Loterie

Temps Réel

## ■ Types d'événements

- ▶ **Périodiques** : distribution vidéo, chaîne industrielle *etc.*
- ▶ **Apériodiques** : monitoring hospitalier, contrôleur de bord *etc.*

## ■ Cas particulier

- ▶ **Systèmes « ordonnancables »** : Soient  $N$  événements périodiques de période  $P_i$ , nécessitant  $C_i$  temps CPU pour s'exécuter. Le système est dit ordonnancable si et seulement si :  $\sum_{i=1}^N \frac{C_i}{P_i} \leq 1$



# Catégories d'ordonnanceurs temps réel

Systèmes  
d'Exploitation

Didier Verna  
EPITA

Généralités

Réquisition  
Critères

Algorithmes

FCFS  
SJF

Priorités  
Tourniquet

Multi-niveau  
Loterie

Temps Réel

## ■ Types d'ordonnanceurs

- ▶ **Temps réel rigide** : « réservation de ressource ». L'ordonnanceur doit connaître exactement les échéances de chaque processus, et les ressources nécessaires.
- ▶ **Temps réel souples** : fournir des priorités hautes et non dégradables, minimiser la latence de dispatching.

## ■ Minimisation de la latence de dispatching

- ▶ Points de réquisition, réquisition des appels systèmes, ou plus généralement de tout le noyau (Solaris 2).

## ■ Inversion des priorités

- ▶ Un processus prioritaire attend des ressources noyau prises par un (des) processus non prioritaire(s)
- ▶ Solution : héritage des priorités (accès noyau en priorité haute).