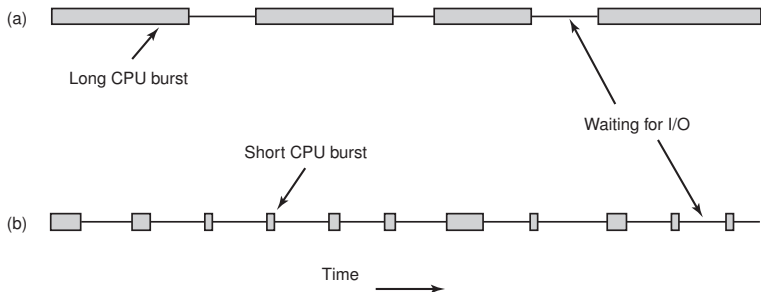




Haute école d'ingénierie et d'architecture Fribourg  
Hochschule für Technik und Architektur Freiburg

# Systèmes d'exploitation

## L'ordonnancement des processus



Des «rafales» (burst) d'utilisation du CPU alternant avec des périodes d'attente d'E/S (entrée/sortie).

- (a) Processus de traitement
- (b) Processus d'E/S



- Lors de la création d'un processus (doit-on continuer avec le parent ou donner le contrôle à l'enfant en premier?)
- Lorsqu'un processus termine
- Lorsqu'un processus se met dans l'état «bloqué» car il doit attendre sur des E/S
- Lors d'une interruption faite par un périphérique (E/S)
- Lors d'une interruption par l'horloge du système (clock) pour les processus préemptifs



«En informatique, la préemption est la capacité d'un système d'exploitation multitâche à exécuter ou stopper une tâche planifiée en cours».



- Traitement par lot (batch). Non-préemptifs, ou alors avec une très longue période.
- Interactifs. Préemptifs avec une courte période.
- Temps réel. Préemption pas nécessaire. Les processus sont connus et collaborent.



**Équité :** attribuer à chaque processus un temps processeur équitable

**Application de la politique :** faire en sorte que la politique définie soit bien appliquée

**Équilibre :** faire en sorte que toutes les parties du système soient occupées



**Capacité de traitement :** optimiser le nombre de jobs à l'heure

**Délai de rotation :** réduire le délai entre la soumission et l'achèvement

**Utilisation du CPU :** faire en sorte que le processeur soit occupé en permanence



**Temps de réponse :** répondre rapidement aux requêtes

**Proportionnalité :** satisfaire aux attentes des utilisateurs





**Respecter les délais :** éviter de perdre des données

**Prévisibilité :** éviter la dégradation de la qualité dans les systèmes multimédias



### **All systems**

Fairness - giving each process a fair share of the CPU

Policy enforcement - seeing that stated policy is carried out

Balance - keeping all parts of the system busy

### **Batch systems**

Throughput - maximize jobs per hour

Turnaround time - minimize time between submission and termination

CPU utilization - keep the CPU busy all the time

### **Interactive systems**

Response time - respond to requests quickly

Proportionality - meet users' expectations

### **Real-time systems**

Meeting deadlines - avoid losing data

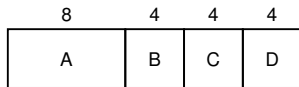
Predictability - avoid quality degradation in multimedia systems



- Premier arrivé, premier servi (first come, first served)
- Exécution du job le plus court en premier (shortest job first)
- Exécution du job suivant avec le temps restant le plus court (shortest remaining time next)



- Pas de préemption



(a)



(b)

Exemple d'exécution du job le plus court en premier (shortest job first) :

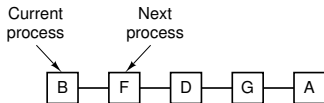
- (a) Exécution des 4 jobs dans leur ordre original
- (b) Exécution du job le plus court en premier
  - On assume que le temps de traitement des processus est connu à l'avance par l'ordonnanceur
  - Le but ici est d'optimiser le délai de rotation si tous les jobs sont soumis en même temps.



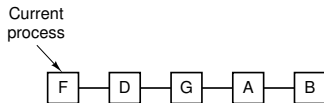
- Algorithme préemptif



- Ordonnancement de type «tourniquet» (round robin scheduling)
- Ordonnancement par priorités (priority scheduling)
- Files d'attente multiples (multiple queues)
- Exécuter le processus suivant le plus court (shortest process next)
- Ordonnancement garanti (guaranteed scheduling)
- Ordonnancement par tirage au sort (lottery scheduling)
- Ordonnancement équitable (fair-share scheduling)



(a)



(b)

Ordonnancement de type «tourniquet» (round robin scheduling).

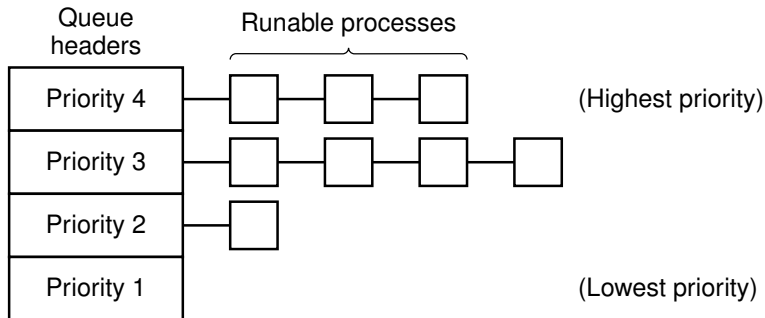
- (a) La liste des processus exécutables
- (b) La liste des processus exécutables après que B a utilisé son quantum



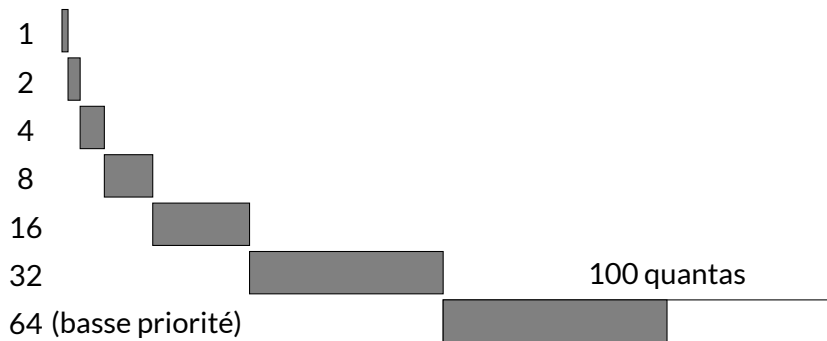


Le «quantum» est l'intervalle de temps pendant lequel un processus est autorisé à s'exécuter. Après avoir épuisé son quantum, le processus doit faire place à un autre processus.

- Un quantum court améliore le temps de réponse
- Un quantum long améliore la performance
- Le quantum est généralement choisi entre 10 mS et 1000 mS
- Les processeurs plus rapides permettent de réduire le quantum



Ordonnement par priorités avec 4 catégories de priorités





- Estimations fondées sur les comportements antérieurs
- Calcul de vieillissement
  - $T_0$
  - $\frac{T_0}{2} + \frac{T_1}{2}$
  - $\frac{T_0}{4} + \frac{T_1}{4} + \frac{T_2}{2}$
  - $\frac{T_0}{8} + \frac{T_1}{8} + \frac{T_2}{4} + \frac{T_3}{2}$



- Si  $n$  processus s'exécutent, chaque processus récupère  $\frac{1}{n}$  cycle du CPU
- Le système effectue le suivi du temps processeur dont chaque processus a bénéficié depuis sa création



- Chaque processus reçoit un ou plusieurs «billets de loterie»
- L'ordonnanceur tire au sort un billet au hasard
- Un processus nouvellement créé pourrait obtenir plus de billets que les anciens processus
- Des processus coopératifs peuvent échanger des billets s'ils le souhaitent



- Cet algorithme tient compte des propriétaires (utilisateurs) des processus
- Chaque **utilisateur** reçoit la même part du CPU
- Si 2 utilisateurs ont droit à 50% du temps de CPU, ils les obtiendront, quel que soit le nombre de processus qu'ils ont démarrés