



Département Informatique

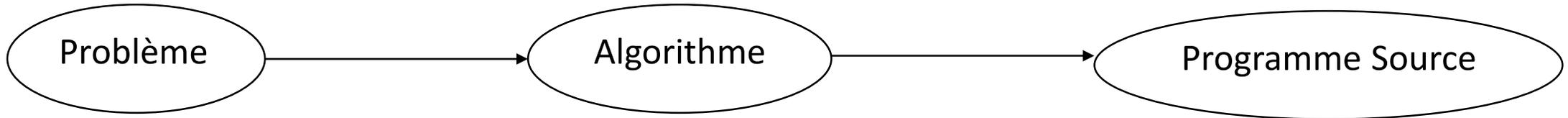
الجمهورية الجزائرية الديمقراطية الشعبية  
République Algérienne Démocratique et Populaire  
وزارة التعليم العالي والبحث العلمي  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
جامعة أحمد زبانة غليزان  
Université Ahmed Zabana - Relizane  
كلية العلوم والتكنولوجيا  
Technologie Faculté des Sciences et de La

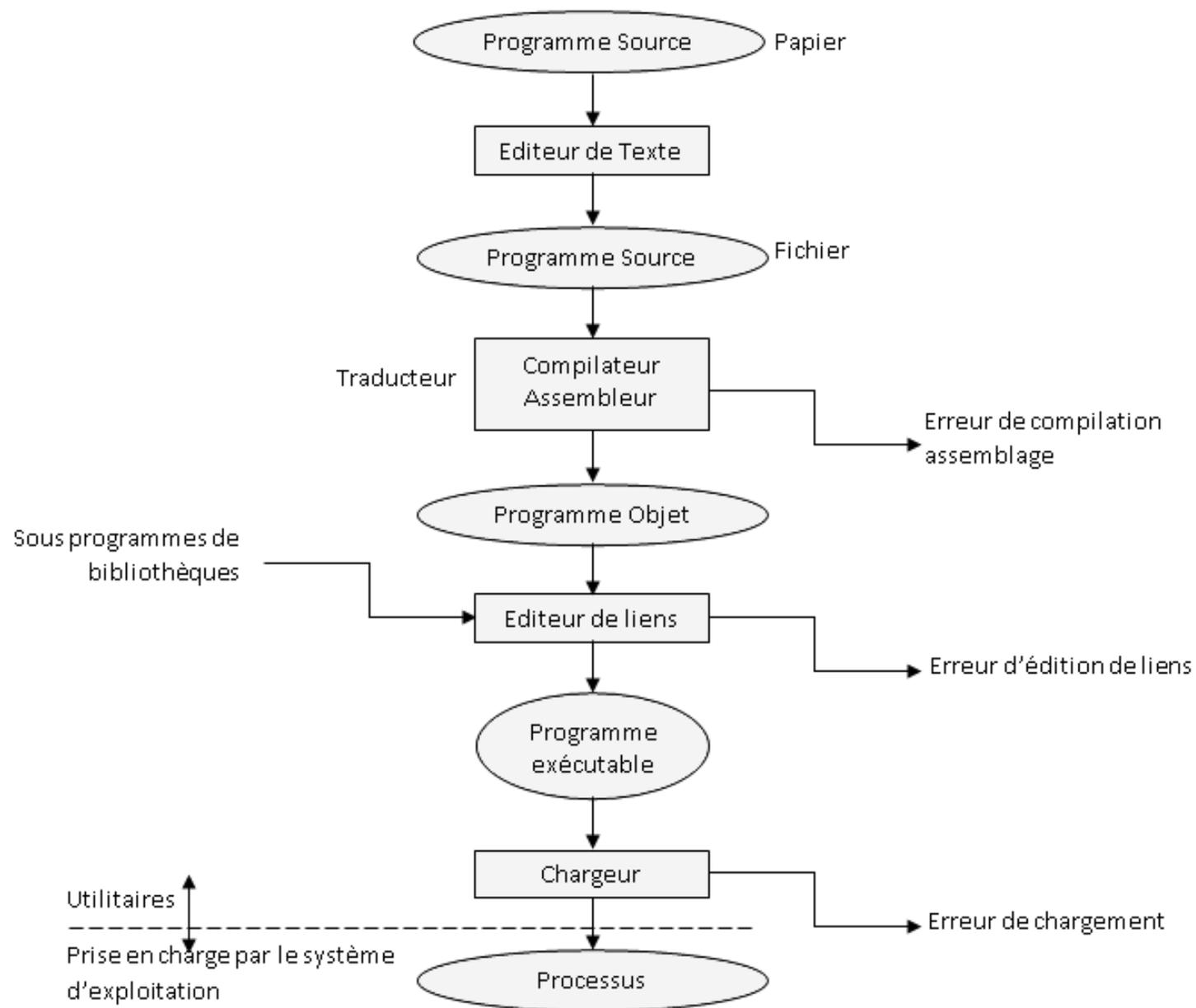
# CHAPITRE 1: Gestion du processeur

Ce cours concerne les L2 Informatique

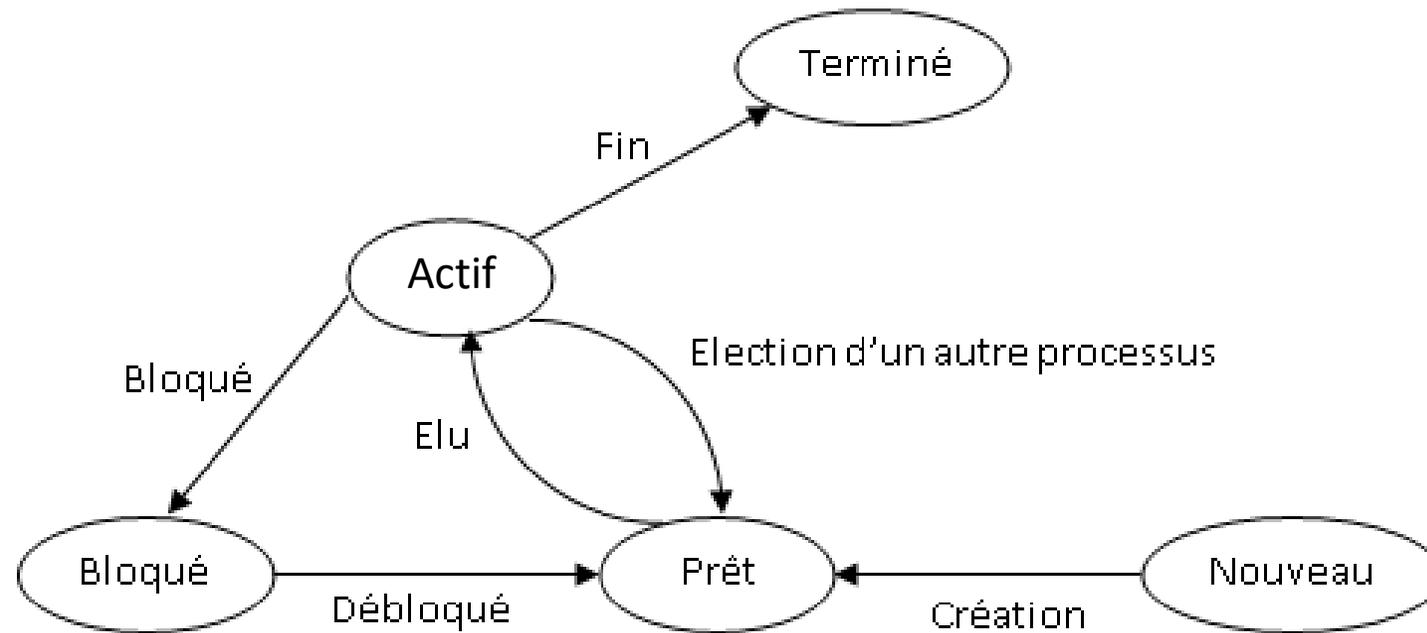
Présenté par: Dr Benotmane. Z

# Cheminement d'un programme dans un système





# Etat de processus



# Ordonnancement (Scheduling)

- Dans un système, plusieurs processus peuvent être présents en mémoire en attente d'exécution. Si plusieurs processus sont prêts, le système d'exploitation doit gérer l'allocation du processeur aux différents processus à exécuter. C'est l'ordonnanceur (Scheduler) qui s'acquitte de cette tâche.
- L'ordonnanceur désigne le composant du noyau du système d'exploitation choisissant l'ordre d'exécution des processus sur les processeurs d'un ordinateur. L'ordonnanceur est appelé scheduler.

# PCB – Process Control Bloc

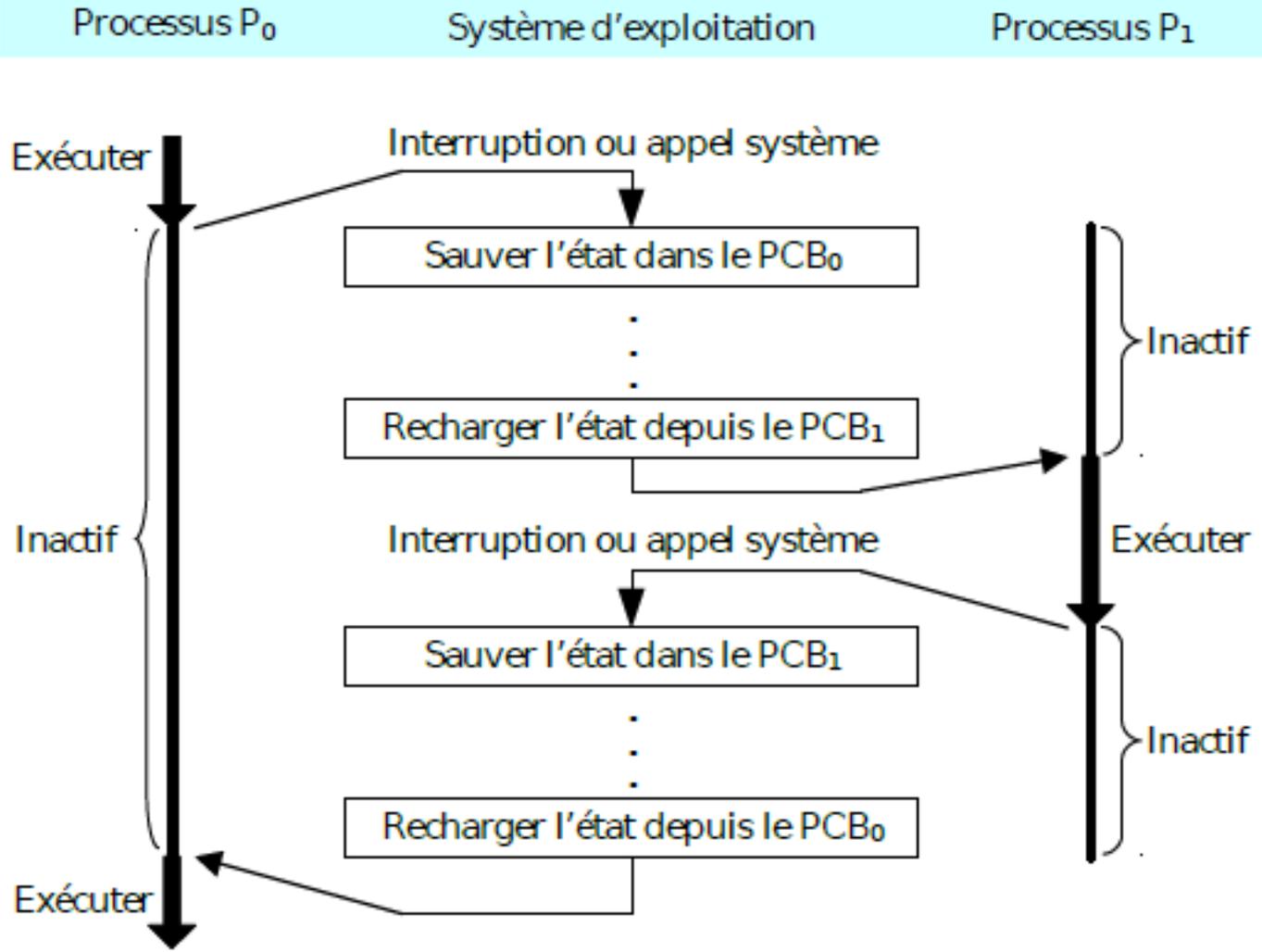
Chaque processus est représenté dans le SE par un **bloc de contrôle de processus**.

ID du processus (pid)
ID du processus père (ppid)
Etat du processus
CO
Registres
Pointeur vers segment de code
Pointeur vers segment de donnée
Pointeur vers segment de la pile
Fichiers ouverts priorité
Signaux
.....

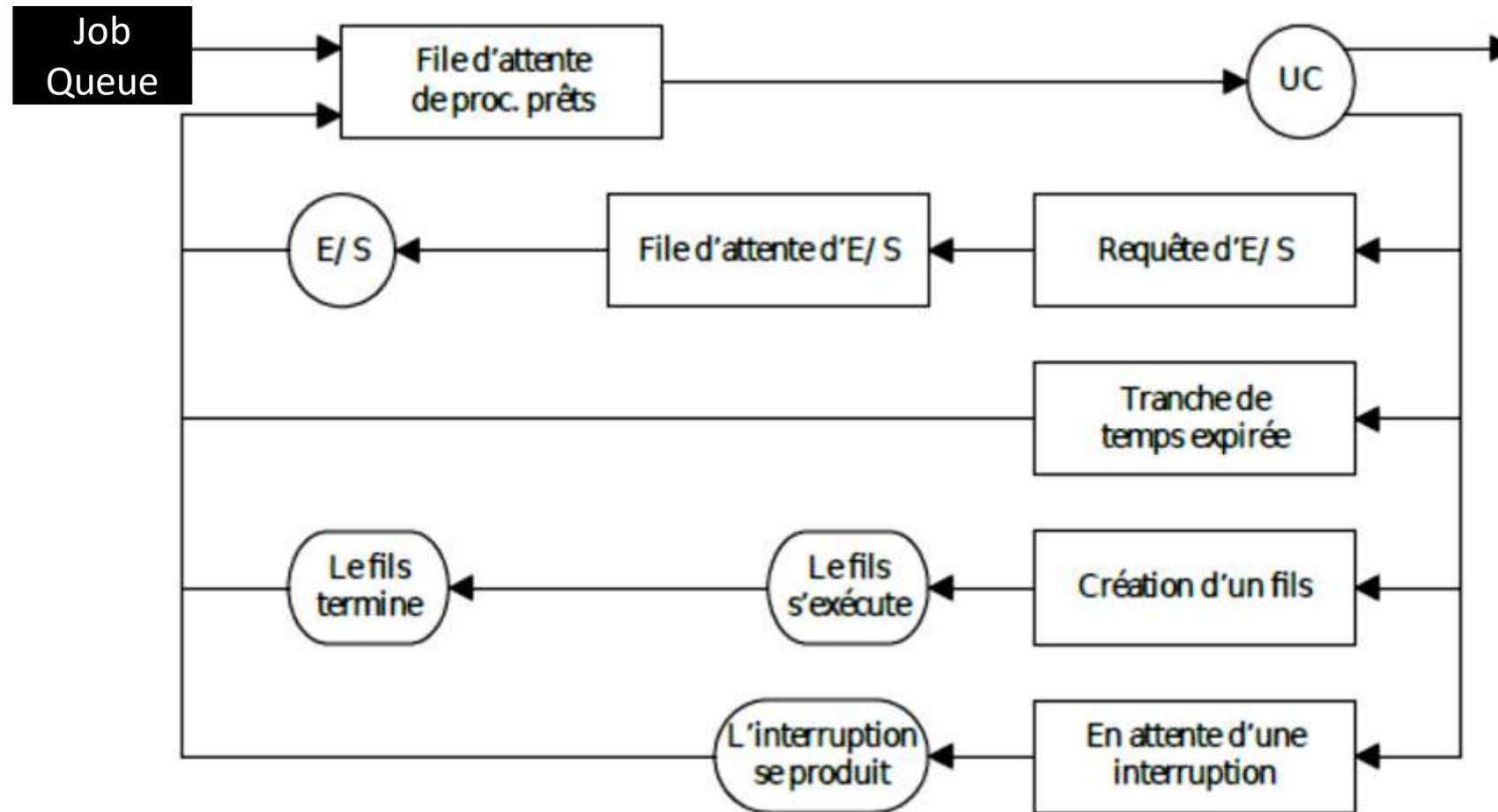
Le système d'exploitation maintient dans une table appelée «**table des processus**» les informations sur tous les processus créés (une entrée par processus : PCB). Cette table permet au SE de localiser et gérer tous les processus.

# Commutation de contexte

- Dans un système multiprogrammé, le processeur assure l'exécution de plusieurs processus en parallèle (pseudo-parallélisme).
- Le passage dans l'exécution d'un processus à un autre nécessite une opération de sauvegarde du contexte du processus arrêté, et le chargement de celui du nouveau processus. Ceci s'appelle la **commutation du contexte (Context Switch)**.
- La commutation de contexte consiste à changer les contenus des registres du processeur central par les informations de contexte du nouveau processus à exécuter.



# Files d'attente du scheduling



# explication

- Quand le processus rentre dans le système, il est initialement inséré dans une **file d'attente de travaux (job queue)**. Cette file d'attente contient tous les processus du système.
- 2. Quand le processus devient résidant sur MC et qui est prêt et attend pour s'exécuter, il est maintenu dans une liste appelée la **file d'attente des processus prêts (Ready queue)**. La file **Prêt** contient donc l'ensemble des processus résidents en MC et prêts à l'exécution.
- 3. Le processus attend dans la file **Prêt** jusqu'à ce qu'il soit sélectionné pour son exécution, et qu'on lui alloue le CPU. Une fois alloué le CPU, le processus devient **Actif**. Actif est concrètement donné par un pointeur (non une file) vers le PCB associé au processus en cours d'exécution.
- 4. Etant actif, le processus pourrait émettre une requête d'E/S vers un périphérique particulier. il est alors placé (son PCB est placé) dans la **file d'attente du périphérique (Device queues)**. Chaque périphérique possède sa propre file d'attente.
- 5. Enfin, le processus actif pourrait créer lui-même un nouveau sous-processus (appelé **processus fils**) et attendre sa fin.

# Critères de scheduling (objectifs)

- **Équité** : attribuer à chaque processus un temps processeur équitable.
- **Temps de réponse**: minimiser le temps de réponse (Temps d'attente).
- **Capacité de traitement**: optimiser le rendement (Débit).
- **Efficacité**: utiliser le processeur à 100%.
- **Etre prédictibles** : évite la dégradation de la quantité dans les systèmes multimédias

# Scheduling avec ou sans préemption

Une politique de Scheduling est dite **non préemptive** si, une fois le processeur central est alloué à un programme, celui-ci le gardera jusqu'à la fin de son exécution, ou jusqu'à se bloquer sur une ressource non disponible ou en attente de la satisfaction d'une demande de ressource.

# Ce qu'on peut calculer..

- Le temps de séjour: pour chaque processus est obtenu en soustrayant le temps d'entrée du processus du temps de terminaison.
- Le temps d'attente: est obtenu en soustrayant le temps d'exécution du temps de séjour.

# Politiques de scheduling :

## 1) FCFS

Le FCFS (Premier Arrivé est le Premier Servi) utilise une file d'attente simple (FIFO).

2) Politique du Job le Plus Court d'Abord (SJF, Shortest Job First) Consiste à servir les processus courts avant les processus longs.

- C'est une politique **non préemptive**, dont la mise en œuvre nécessite la connaissance préalable du temps d'exécution des processus.

3) Politique du Job ayant le Plus Court Temps Restant (SRTF, Shortest Remaining Time First)

lorsqu'un processus est en cours d'exécution, et qu'un nouveau processus ayant un temps d'exécution plus court que celui qui reste pour terminer l'exécution du processus en cours, ce processus est arrêté (préempté), et le nouveau processus est exécuté. Cette méthode équivaut à la méthode SJF mais **préemptive**.

#### 4) Politique à base de priorité

Des priorités sont affectées aux différents processus et ils sont activés en fonction de cette priorité. Le processus élu par le scheduler est celui qui a la plus haute priorité parmi les processus éligibles.

- La priorité affectée à un processeur peut dépendre de l'utilisateur qui a lancé l'exécution du processus, de la quantité de ressources demandée par le processus, etc. Dans le cas d'une politique à **base de priorité préemptive**, l'arrivée d'un processus plus prioritaire entraîne l'arrêt du processus en cours d'exécution et l'attribution du processeur au nouveau processus.
- Cette politique peut poser le problème de **famine**, où les processus moins prioritaires risquent de ne jamais être exécutés. Une solution à ce problème est l'utilisation du principe de **vieillessement** qui permet de modifier la priorité d'un processus en fonction de son âge et de son historique d'exécution.

## 5) Politique du Tourniquet (Round Robin, RR)

- Cette politique consiste à allouer le processeur aux processus suivant une durée d'exécution limitée appelée « **Quantum** ». C'est une politique **préemptive**, qui s'adapte bien aux systèmes à temps partagé.
- **Implantation de l'algorithme Round Robin** Elle est réalisée à l'aide d'une file d'attente circulaire des processus prêts, organisée en FIFO. Un processus qui arrive est mis en queue de la file. Le processus élu par le Scheduler est celui en tête de file.
- Pour signaler l'écoulement du Quantum, on utilise une horloge, initialisée au lancement du processus élu. Celui-ci se terminera de deux manières :
- - Soit à l'écoulement du Quantum, une interruption horloge se déclenche et le processus est remis alors en queue de la file, après avoir sauvegardé le contexte du processus dans son PCB
- - Soit par libération du processeur à la suite d'une demande de ressource ou de fin totale d'exécution.
- Les performances de cette politique dépendent de :

1) **La valeur du Quantum:** Si le Quantum est très grand, la politique Round Robin se confond avec la politique FCFS. Si le Quantum est très petit, les processus auront l'impression de disposer du processeur à lui seul. En effet, le processeur tourne à la vitesse de  $1/n$  de sa vitesse réelle ( $n$  étant le nombre de processus)

2) **La durée de commutation:** Cette commutation se fait à chaque allocation du processeur et nécessite un certain temps d'exécution.

# Politique à Plusieurs Niveaux de Queues (Multi-Level Queues)

- Dans cette stratégie, la file des processus prêts est subdivisée en plusieurs files suivant la classe des processus (batch, interactifs, temps réel, ...).
- Chaque file est gérée suivant une politique de Scheduling propre à elle, et s'adaptant mieux à la classe des processus qu'elle contient.

Exemple:

Processus batch → FCFS.

Processus interactifs → Round Robin.

Le Scheduling entre les files elles-mêmes peut se faire :

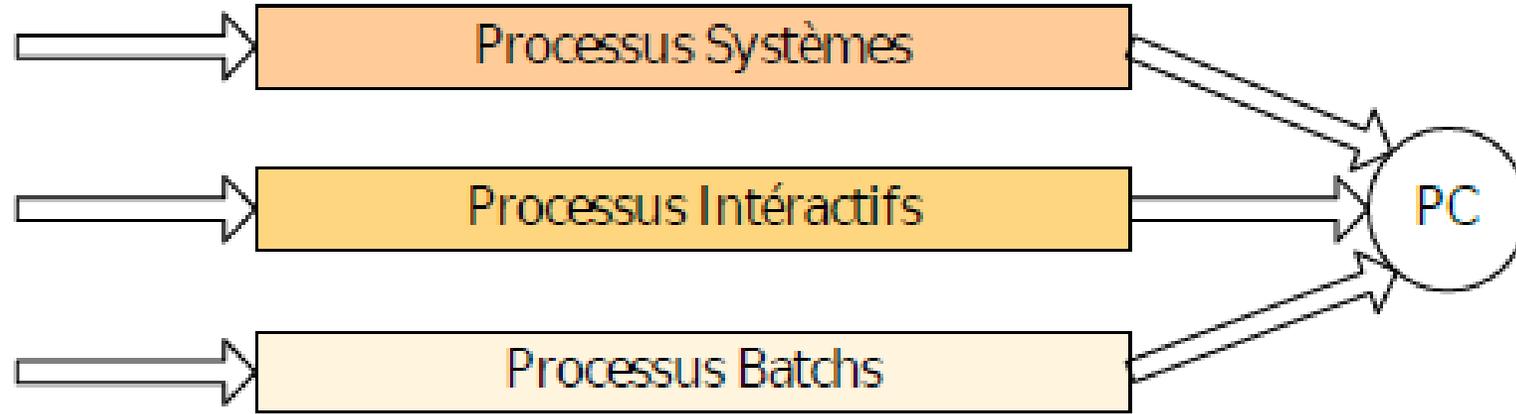
Soit par priorité.

**Ex.** la file INTERACTIF est plus prioritaire que la file Batch.

Soit par partage du temps du processeur.

**Ex.** 80% → Interactifs, 20% → Batch.

Plus haute priorité



Plus basse priorité

# Politique à Plusieurs Niveaux de Queues Dépendantes (Multilevel Feedback Queues)

Dans la politique précédente, un processus ne change pas de files. Par contre, dans la politique à plusieurs niveaux dépendants, un processus peut changer de file suivant les changements de comportement qu'il peut avoir durant son exécution.

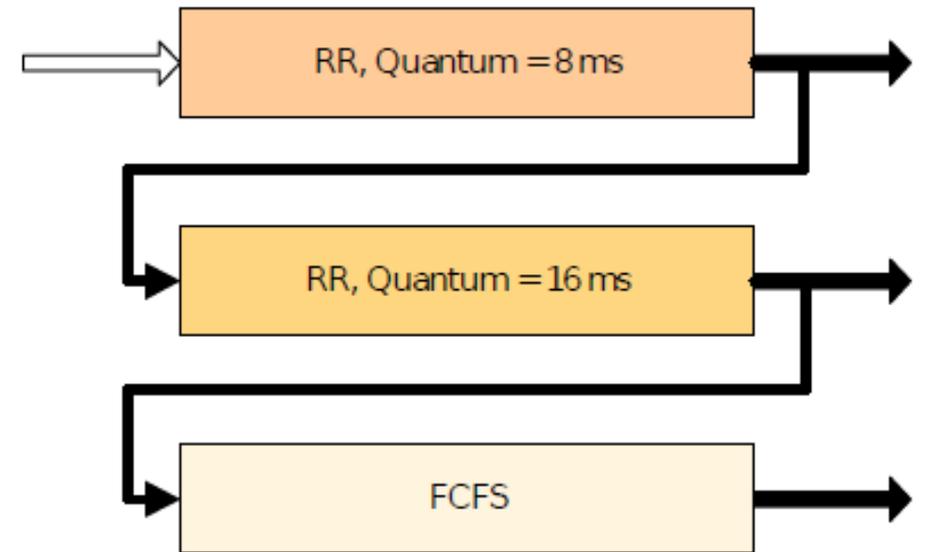
Ceci est réalisé afin d'isoler les processus consommateurs de temps CPU, et pouvoir lancer des processus de faible priorité en les plaçant dans des files à plus haute priorité. En effet :

- Un processus fait parfois beaucoup de calcul et parfois beaucoup d'E/S.
- Un processus qui a séjourné un grand temps dans une file à haute priorité est placé dans une file de plus basse priorité.

Cette politique est définie par les paramètres suivants :

- Nombre de files.
- L'algorithme de Scheduling de chaque file.
- Une méthode déterminant dans quelle file placer un nouveau processus.
- Une méthode de transition d'un processus d'une file à une autre.

# exemple



Soient Trois files:

- Q0 – gérer par la politique RR avec un quantum de 8 millisecondes.
- Q1 – gérer par la politique RR avec un quantum de 16 millisecondes.
- Q2 – gérer par la politique FCFS.

La politique de scheduling appliquée est décrite comme suit :

- Un nouveau processus est placé dans Q0 au début; à sa première exécution, il reçoit 8 millisecondes. S'il ne termine pas son exécution, il est replacé dans Q1.
- Si un processus de la file Q1 est servi (16 msec) et ne se termine pas, il est replacé dans Q2.

# Les threads

Un thread (processus léger) offre la possibilité d'effectuer plusieurs tâches au même moment. Le processus est alors divisé en plusieurs flots d'exécution.

Un thread partage son code et ses données avec d'autres threads du même processus.

Exemple un navigateur web peut avoir un thread qui affiche des images, et un autre qui récupère les données du réseau.