

Gestion des processus

Qu'est-ce qu'un processus?

- **Définition :**

Un processus est un objet dynamique qui fournit à un instant donné l'image de l'état d'avancement de l'exécution sur un processeur d'un programme produit par compilation et édition de lien .

- Différence entre processus et programme :
le programme est une description statique

le processus est une activité dynamique (il a un début, un déroulement et une fin, il a un état qui évolue au cours du temps)

Une analogie pourrait aider à mieux comprendre la différence entre processus et programme:

Un boulanger en train de confectionner un gâteau : Il dispose d'une cuisine équipée, il a la recette et tous les ingrédients nécessaires : farine, œufs, sucre, ...

- La recette = Le programme
- Le boulanger = Le processeur
- Les ingrédients = Les arguments, paramètres d'entrés
- **Le processus** est l'activité qui consiste à confectionner le gâteau: lecture de la recette, mélange des ingrédients et cuisson.

2.2) Les différents états d'un processus

Un processus peut être dans 3 états possibles :

- élu: En cours d'exécution.
- prêt: Prêt à être élu.
- bloqué: Attente d'une autre ressource autre que le processeur

Schéma cas non préemptif :

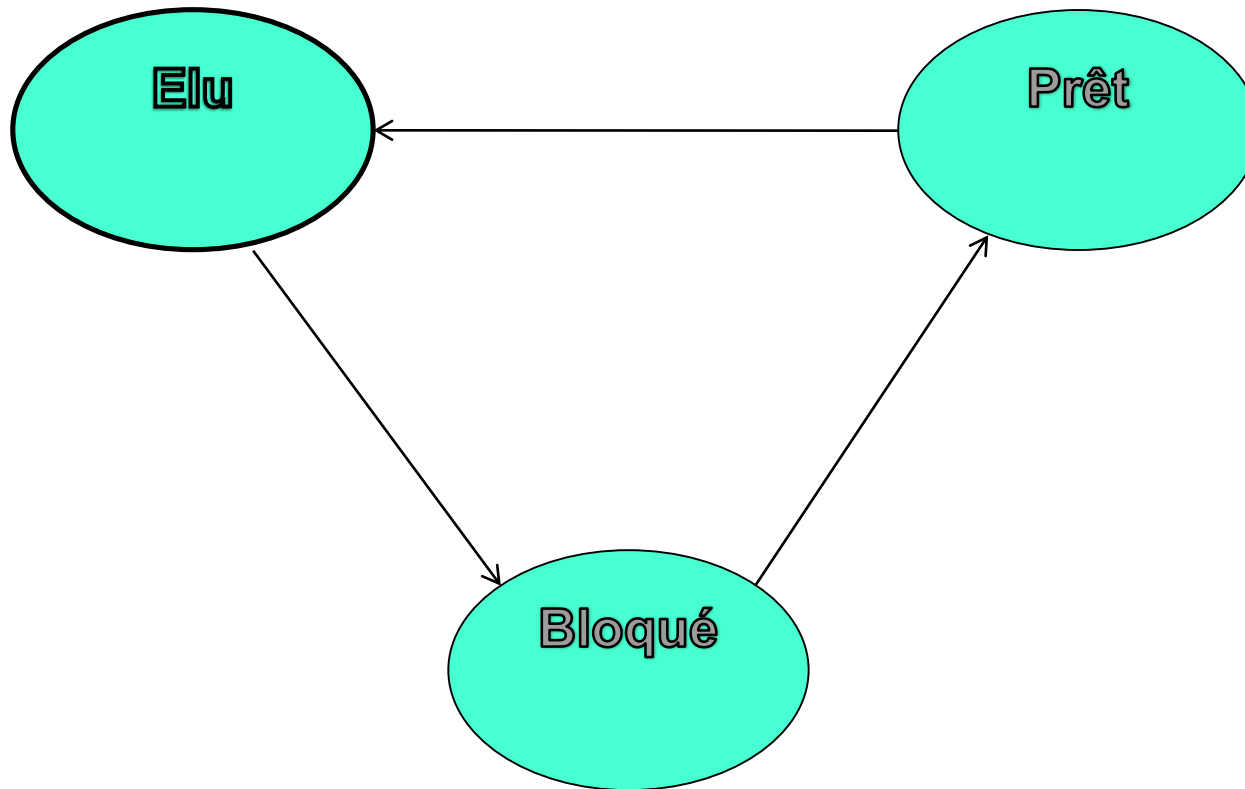
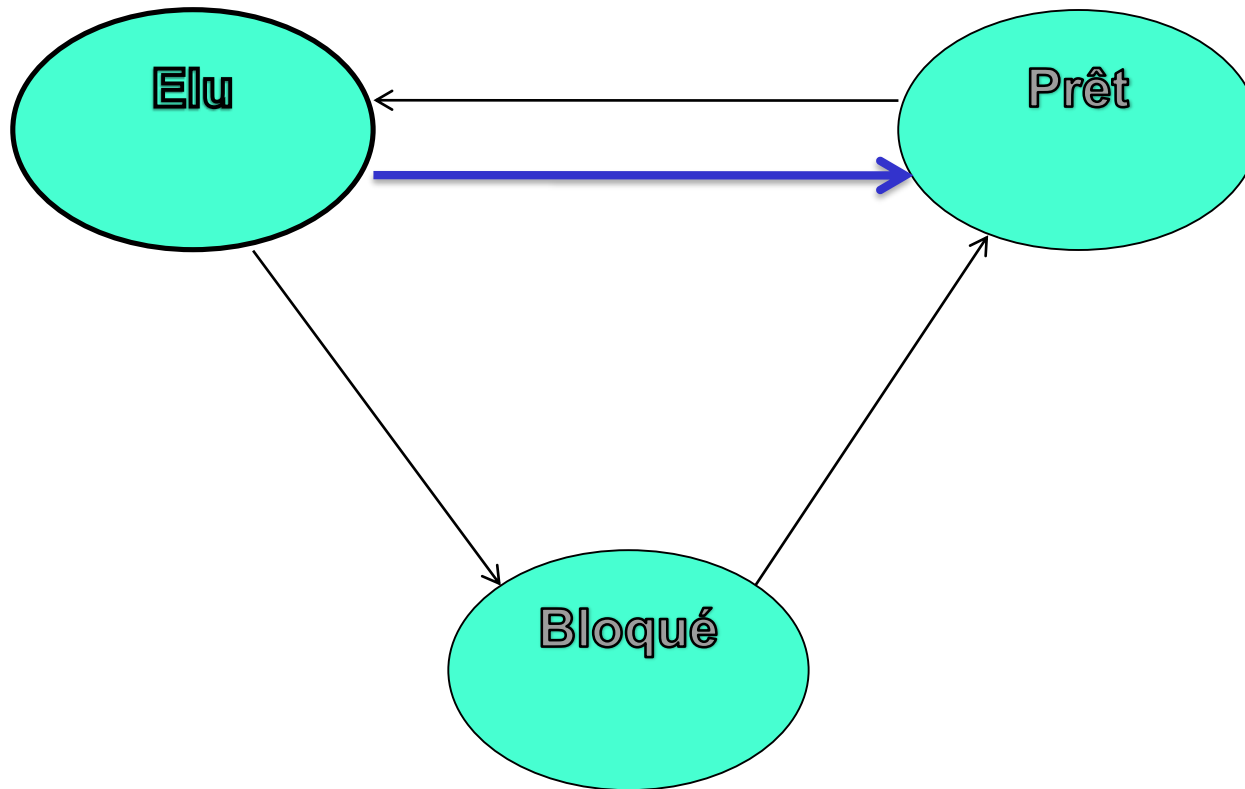


Schéma cas préemptif :



La signification des 4 transitions légales est la suivante :

Prêt → Elu: L'ordonnanceur élit ce processus parmi les processus prêts

Elu → Prêt: Le processus a épuisé le quantum de temps qui lui a été attribué → élection d'un autre processus

Elu → Bloqué: Bloqué en attente d'un événement externe
(données...) → L'ordonnanceur averti par le
processus bloqué → élection d'un autre
processus

Bloqué → Prêt: L'événement attendu se produit →
traitement par le SE → le SE fait passer le
processus de l'état bloqué à prêt

Exemple : Fin d'une lecture dans un fichier

Ordonnancement des processus

- Plusieurs processus sont prêts à être exécutés → le SE doit faire un choix.
- La partie du SE qui effectue ce choix est **l'ordonnanceur (scheduler)**
- L' algorithme que l'Ordonnanceur utilise s'appelle **algorithme d'ordonnancement**

Rôle de l'ordonnanceur:

L'Ordonnanceur est un programme intégré au noyau

L'Ordonnanceur (scheduler) définit l'ordre dans lequel les processus prêts utilisent l'UC et la durée d'utilisation, en utilisant un algorithme d'ordonnancement.

Algorithmes

Il existe deux familles d'algorithmes :

- **Sans réquisition (ASR) = non préemptif** : un processus est exécuté jusqu'à la fin

Le choix d'un nouveau processus ne se fait que sur blocage ou terminaison du processus courant (utilisé en batch par exemple)

inefficace et dangereux (ex: exécution d'une boucle sans fin...)

- **Avec réquisition (AAR) = préemptif** : à intervalle régulier, l'ordonnanceur reprend la main et élit un nouveau processus actif

Définition : On appelle le **temps de rotation** d'un processus l'intervalle de temps séparant la date de la **soumission** de la date de **fin** d'exécution.

Définition : On appelle le **temps de réponse** d'un processus l'intervalle de temps séparant la date de la **soumission** de la date du début d'exécution.

Définition : On appelle le **temps moyen** d'exécution d'un ensemble de tâches la moyenne des intervalles de temps séparant la **soumission** d'une tâche de sa **fin** d'exécution.

Exemple: soient 4 processus A,B,C et D arrivés au même moment avec des temps d'exécution respectifs de 8,4,6,2 minutes, exécuté dans cet ordre, le temps de rotation (soumission → fin) de A est 8, celui de B est 12, celui de C est 18 et celui de D est 20,

pour un temps moyen de **$(8+12+18+20) / 4$**

ASR

- Ordonnancement dans l'ordre d'arrivée
premier arrivé, premier servi = FIFO Le prochain processus élu est le plus ancien dans la file d'attente .

Cet algorithme est facile à comprendre et à programmer

- PCTE = Plus Court Temps d'Exécution = exécution du job le plus court en premier :

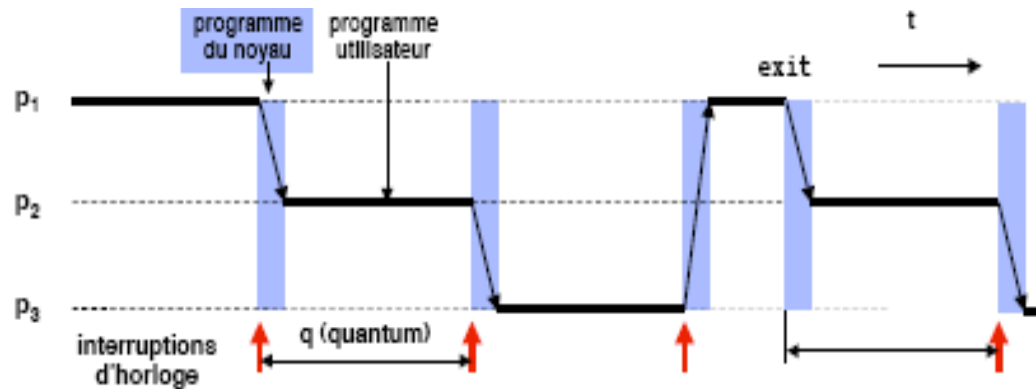
Le prochain processus élu est le plus court.

Cet algorithme suppose que les délais d'exécution soient connus par avance,

AAR = Algorithme avec réquisition

A chaque signal d'horloge, le SE reprend la main, décide si le processus courant a consommé son quota de temps machine et alloue éventuellement le processeur à un autre processus

Allocation du processeur aux processus



Le processeur est alloué par tranches de temps successives (quanta) aux processus prêts à s'exécuter (non bloqués).

- Dans la pratique, la valeur du quantum est d'environ 20 à 50 ms
- La commutation entre processus prend un temps non nul (Elle est réalisée par l'ordonnanceur (scheduler))

Mécanisme d'allocation du processeur = Mécanismes de commutation

Que fait précisément le noyau du système lors de la commutation de processus ?

1. Déterminer le prochain processus élu . Le mécanisme utilise une file d'attente définie selon l'algorithme.

2. Réaliser l'allocation proprement dite. Deux étapes :

a) Sauvegarder le contexte du processus élu actuel (les données, variables, PID, User ...), pour pouvoir le retrouver ultérieurement, en vue de la prochaine allocation

b) Restaurer le contexte du nouveau processus élu puis lancer son exécution.

– il existe de nombreux algorithmes d'ordonnancement avec réquisition :

Ordonnancement circulaire (tourniquet)

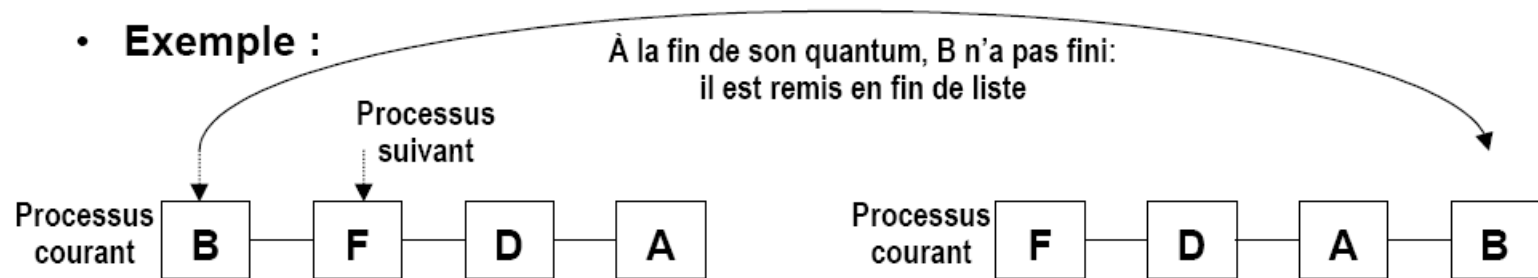
Il s'agit d'un algorithme ancien, simple et fiable.

Chaque processus dispose d'un quantum de temps pendant lequel il est autorisé à s'exécuter.

Si le processus actif se bloque ou s'achève avant la fin de son quantum, le processeur est immédiatement alloué à un autre processus.

Si le quantum s'achève avant la fin du processus, le processeur est alloué au processus le plus ancien dans la liste d'attente et le processus précédent se trouve ainsi en queue de liste.

- **Exemple :**



Problème = réglage du quantum :

– quantum trop petit → beaucoup de commutations (= temps de passage d'un processus à l'autre) : le processeur passe son temps à commuter

Avantage: *un petit quantum assure un temps de réponse rapide*

– quantum trop grand : augmentation du temps de réponse d'une commande (même simple)

Avantage: un grand quantum minimise le temps de changement de contexte

Ordonnancement avec priorité

Le modèle du tourniquet suppose que tous les processus sont d'égale importance. C'est irréaliste.

D'où l'attribution de priorité à chaque processus.

L'ordonnanceur lance le processus prêt de priorité la plus élevée.

Pour empêcher le processus de priorité la plus élevée d'accaparer le processeur, l'ordonnanceur abaisse à chaque interruption d'horloge la priorité du processus actif.

Deux choix sont possibles pour avoir la commutation:

1) Si la priorité du processus actif devient inférieure à celle du deuxième processus de plus haute priorité, la commutation a lieu.

Ou bien

2) Assigner un quantum à chaque processus autorisé à s'exécuter . Lorsque ce quantum est écoulé, le processus prioritaire suivant peut enfin s'exécuter.