

9.3

Ordonnancement

NSI TERMINALE - JB DUTHOIT

Dans un système multitâche, plusieurs processus sont actifs simultanément, mais un processeur (simple cœur) ne peut exécuter qu'une instruction à la fois !

☞ Nous allons donc avoir l'impression que le processeur gère les processus de façon simultanée, mais il en est rien.

Il va donc falloir partager le temps de processeur disponible entre tous les processus : c'est le travail de **l'ordonnanceur**. Ce dernier a pour tâche de sélectionner le processus suivant à exécuter parmi ceux qui sont prêts.

Définition

L'ordonnanceur gère les processus, ayant pour objectif de partager le temps d'utilisation du processeur. Généralement, l'ordonnanceur applique une politique de temps partagé. L'algorithme d'ordonnancement choisit un processus en attente d'exécution (en fonction de différents critères liés au système d'exploitation). Le processus élu s'exécute alors pendant un certain temps, appelé **quantum** d'ordonnancement. A la fin de ce quantum, l'ordonnanceur interrompt le processus et en choisit un nouveau.

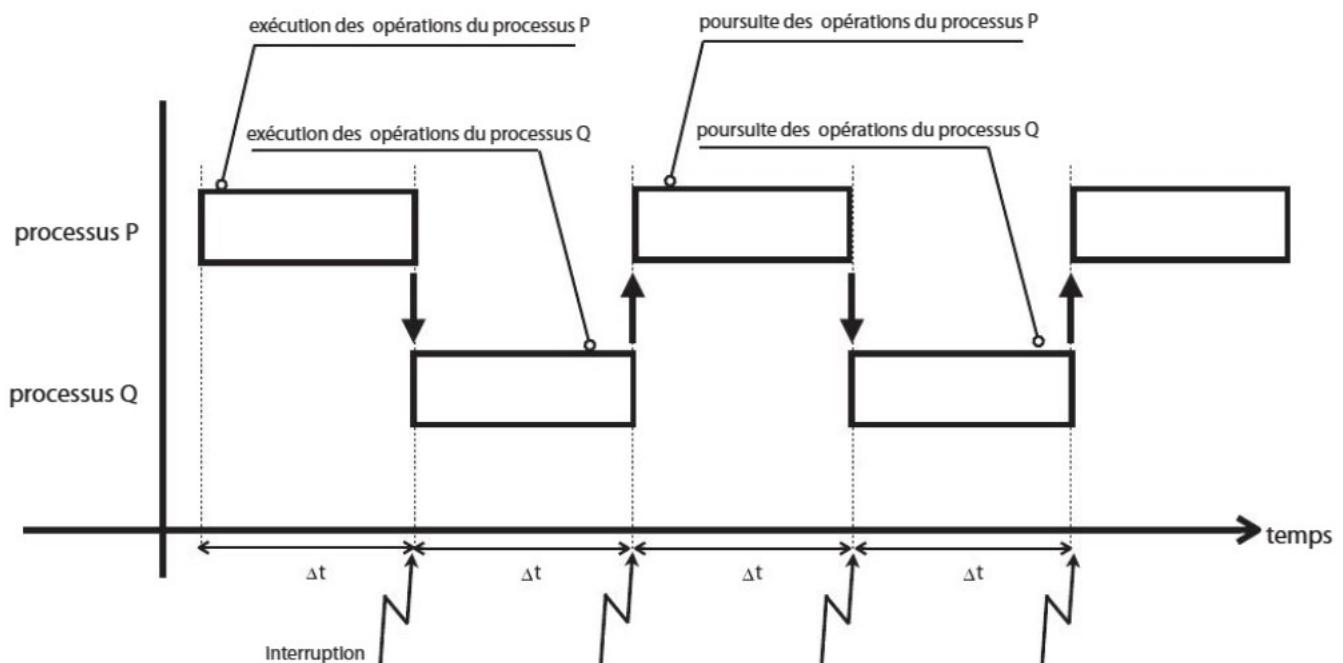


Illustration par un chronogramme de l'ordonnancement de 2 processus

9.3.1 Ressources

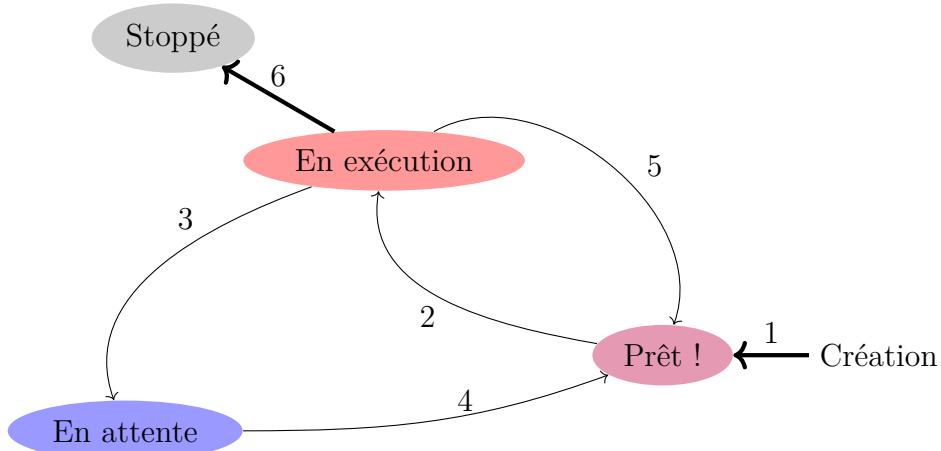
Durant son exécution, le processus peut avoir besoin de ressources matérielles (disque dur, carte réseau...) ou de ressources logicielles.

A un instant donné, une ressource peut être **libre** (aucun processus n'utilise cette ressource) ou **occupé** (un processus utilise la ressource).

9.3.2 État d'un processus

Afin de gérer cet ordonnancement, les processus se voient affectés d'états différents :

- prêt (ready) : le processus attend son tour pour prendre la main
- (running) : le processus a accès au processeur pour exécuter ses instructions
- (sleeping) : le processus attend qu'un événement se produise (saisie clavier, réception d'une donnée par le réseau ou le disque dur ...) en exécution .
- (stopped) : le processus a fini son travail ou a reçu un signal de terminaison (SIGTERM, SIGKILL, ...). Il libère les ressources qu'il occupe.



1 Le processus est créé ; il est "prêt"

2 **Élection** L'ordonnanceur choisit ce processus. Il entre en exécution.

3 Le processus se met en attente d'un événement.

4 L'événement attendu se produit. Le processus est donc "prêt"

5 **préemption** : L'ordonnanceur décide de suspendre le processus afin de donner la main à un autre processus.

6 Le processus est terminé.

☞ Il est vraiment important de bien comprendre que c'est le système d'exploitation qui attribue aux processus les différents états "En exécution, En attente, prêts".

On dit que le système d'exploitation gère l'ordonnancement des processus (tel processus sera prioritaire sur tel autre...)

Afin d'élire le processus qui va repasser en mode exécution, l'ordonnanceur applique un algorithme prédefini lors de la conception de l'OS.

Le choix de cet algorithme va impacter directement la réactivité du système et les usages qui pourront en être fait. C'est un élément critique du système d'exploitation.

Parmi les algorithmes les plus répandus :

- Le modèle Round Robin (**tourniquet**) : La ressource est affectée à chaque processus à tour de rôle. Cette méthode a plusieurs avantages : simplicité, rapidité de gestion, robustesse.
- Mise en place d'un système de priorité : l'ordre de l'affectation de la ressource sera alors fonction de la priorité des tâches.
⚠️ Attention au réglage du niveau de priorité, qui doit être "équitable" et "objectif".
- Gestion du premier entrée, premier sortie (FIFO). Comme par exemple la file d'impression des documents sur une imprimante.
- SRTF(Shortest Remaining Time First) ou SJF (Shortest Job First) : Le plus court d'abord. Assez efficace, mais la difficulté réside dans le fait d'estimer la durée du processus en amont, avant qu'il ne commence.

Exercice 9.3

Considérons trois processus notés P1,P2 et P3, dont les temps d'exécution et leurs arrivages respectifs sont donnés dans le tableau ci-dessous.

Processus	P1	P2	P3
Durée (en quantum)	8	5	9
Arrivée	8	3	0

Construire le schéma d'ordonnancement suivant le modèle FIFO, "premier arrivé, premier servi" ou "Le premier d'abord". ⚠️ Cet ordonnancement est non préemption

Exercice 9.4

Considérons quatre processus notés P1,P2,P3 et P4, dont les temps d'exécution et leurs arrivages respectifs sont donnés dans le tableau ci-dessous.

Processus	P1	P2	P3	P4
Durée (en quantum)	8	5	9	2
Arrivée	4	0	3	7

Construire le schéma d'ordonnancement suivant le modèle "Le plus court d'abord"

Exercice 9.5

Considérons 3 processus notés P1,P2 et P3, dont les temps d'exécution et leurs arrivages respectifs sont donnés dans le tableau ci-dessous.

Processus	P1	P2	P3
Durée (en quantum)	8	5	9
Arrivée	1	0	3

Construire le schéma d'ordonnancement suivant le modèle "du tourniquet". Parmi l'ensemble des processus prêts, le système choisit celui qui n'a pas été exécuté depuis le plus longtemps.

Exercice 9.6

Considérons cinq processus notés P1,P2,P3,P4 et P5 dont les temps d'exécution et leurs arrivages respectifs sont donnés dans le tableau ci-dessous.

Processus	P1	P2	P3	P4	P5
Durée (en quantum)	3	6	4	2	1
Arrivée	0	1	4	6	7
Temps de terminaison					
Temps de séjour					
Temps d'attente					

Répondre aux questions suivantes :

1. Construire le schéma d'ordonnancement suivant le modèle "Le plus court en premier"
2. Compléter le tableau

Définition

Le **temps de terminaison** d'un processus est la durée écoulée entre le temps 0 et le temps où le processus est terminé.

Définition

Le **temps de séjour (ou temps d'exécution)** d'un processus est la différence du temps d'arrivée et du temps de terminaison.

Définition

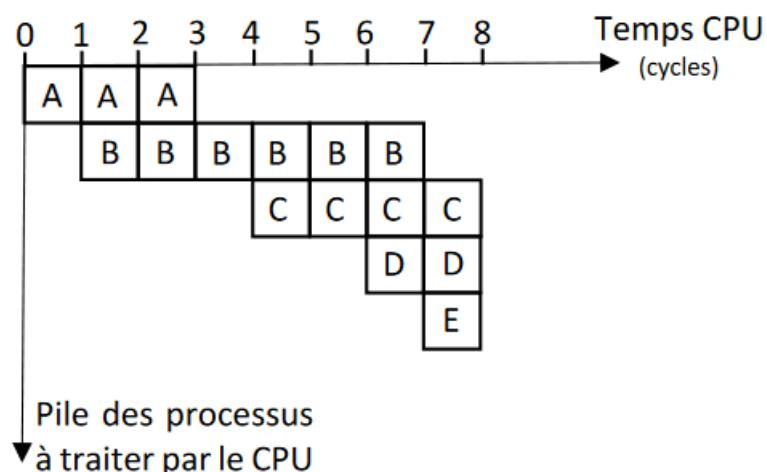
Le **temps d'attente** d'un processus est la différence entre le temps de séjour et la durée du processus.

Exercice 9.7

Considérons cinq processus notés A, B, C, D et E, dont les temps d'exécution et leurs arrivages respectifs sont donnés dans le tableau ci-contre.

Processus	Temps d'exécution	Temps d'arrivée
A	3	0
B	6	1
C	4	4
D	2	6
E	1	7

On peut également représenter le tableau précédent de la manière suivante afin de faciliter la compréhension de l'ordonnancement des processus :



Représenter pour chaque algorithme le schéma d'exécution. Calculer le temps de séjours, le temps moyen de séjour, le temps d'attente (soustraction entre le temps de séjour et le temps d'exécution) et enfin le temps d'attente moyen

1. Algorithme "Premier arrivé, premier servi"
2. Algorithme "Le plus court avant".
3. Algorithme du tourniquet

Exercice 9.8

Un ordonnancement préemptif de quantum $q = 2$ unités est mis en œuvre pour gérer des processus qui se partagent une ressource R utilisée en exclusion mutuelle.
Un processus A est lancé au temps $t = 0$, son exécution nécessite $d = 9$ unités de temps. De plus, ce processus utilise la ressource R à partir de son temps d'exécution 3 pour une durée de 5 unités de temps. Des processus B et C sont créés aux temps 3 et nécessitent respectivement 6 et 4 unités de temps. Le processus B utilise la ressource R durant 4 unités de temps après une durée d'exécution d'une unité de temps.

Indiquer quel est le processus élu à chaque instant.