

Systèmes d'exploitation TD 1 : Gestion des processus (Ordonnancement)

Enseignante : Fadoua DRIRA HAMMAMI

Exercice 1 :

1. Laquelle de ces affirmations décrit le mieux la relation entre programme et processus ?
☐ un programme exécute un processus ☐ un processus exécute un programme
☐ un programme comporte plusieurs processus ☐ un processus comporte plusieurs programmes
2. Quand un appel système se produit-il ?
☐ lorsque le système d'exploitation démarre ☐ lorsqu'un processus souhaite utiliser une fonction du système d'exploitation ☐ lorsqu'un périphérique émet une interruption ☐ lorsque qu'un processus dépasse son quantum de temps
3. L'Ordonnanceur du noyau d'un système d'exploitation :
☐ n'aide à gérer qu'une seule tâche à la fois ☐ s'appuie sur les notions de commutation de contexte et d'ordonnancement. ☐ fait basculer le processus entre ses trois états : élu, prêt, bloqué
4. Définir brièvement la commutation de contexte et dans quel cas elle peut avoir lieu ?
5. Laquelle de ces transitions n'est pas possible pour un processus ? Justifier.
☐ Bloqué → Prêt ☐ Prêt → Bloqué ☐ Bloqué → Élu ☐ Élu → Prêt
6. Un système d'exploitation peut-il s'installer sur n'importe quelle machine ? Justifier
7. Vous allumez l'ordinateur, qu'est-ce qui se passe au niveau du système d'exploitation ?
8. Quelles sont les inconvénients des systèmes mono-tâche ?
9. Est-ce qu'un SE multitâche est nécessairement multi-utilisateur ? Et pour l'inverse ? Expliquez.
10. Pour le choix du quantum, donnez un inconvénient pour un quantum trop petit et pour un très élevé.

11. Expliquez la différence entre les Ordonnanceurs d'exécution non préemptifs (ex : ceux des premières versions de MacOS) et préemptifs (ex. ceux d'Unix, de Windows2000...). Que se passe-t-il pour les processus qui sont à l'état prêt dans les deux cas ?

Exercice 2 :

On considère les tâches T1, T2, T3 et T4 dont chacune dispose d'une priorité qu'elle conserve tout au long de son exécution. On suppose qu'il existe 4 niveaux de priorités numérotées de 0 à 3 (0 étant la plus faible priorité).

	Temps CPU	E/S	Durée E/S	Priorité
T1	300 ms	aucune		0
T2	20ms	Toutes les 10ms	250ms	3
T3	200ms	aucune		1
T4	40ms	Toutes les 20ms	180ms	3

Pour un quantum de temps de 100 ms et sachant que, pour les tâches de même priorité, les tâches les plus longues sont défavorisées :

- Décrivez l'évolution du système : instant, nature de l'événement (commutation (nom_de_la_tâche_associée) et/ou demande d'E/S (nom_de_la_tâche_associée) et/ou fin d'E/S (nom_de_la_tâche_associée) et/ou fin (nom_de_la_tâche_associée)), tâche élue.
- Donnez pour chaque tâche l'instant au bout duquel elle termine son exécution.

Exercice 3 :

On considère 4 processus, A, B, C, D. On suppose que l'exécution des processus nécessite :

- ☐ Pour A : 7 unités de temps CPU, 3 unités de temps d'E/S et 5 unités de temps CPU.
- ☐ Pour B : 6 unités de temps CPU, 4 unités de temps d'E/S, 4 unités de temps CPU.
- ☐ Pour C : 5 unités de temps CPU.
- ☐ Pour D : 1 unité de temps CPU, 4 unités de temps d'E/S et 2 unités de temps CPU.

On suppose que :

- ☐ A se présente en premier, à l'instant 0,
- ☐ B se présente à l'instant 1,
- ☐ C se présente à l'instant 9,
- ☐ D se présente à l'instant 12.

Montrez comment les 4 processus vont utiliser le processeur dans chacun des cas suivants:

- Chaque processus a son propre périphérique d'E/S et l'ordonnanceur fonctionne selon Premier Arrivée Premier Servi PAPS (sans préemption).
- Chaque processus a son propre périphérique d'E/S et l'ordonnanceur utilise l'algorithme du tourniquet, avec un quantum de 5. Le temps de commutation est égal à 0. Donnez, dans ce cas, les temps de séjour des processus A, B, C et D.

3) Les trois processus utilisent le même périphérique d'E/S dont la file d'attente est gérée premier arrivée premier servi. L'ordonnanceur du processeur utilise l'algorithme du tourniquet, avec un quantum de 5.

Exercice 4 :

A l'instant t , deux processus utilisateur P1 et P2 existent dans un système monoprocesseur. Le processus P1 est composé de deux threads. Le processus P2 est composé de trois threads. Les temps nécessaires à leurs exécutions sont résumés dans le tableau suivant :

Proc.	Threads	Temps d'exécution
P1	T11	1 unité de CPU
	T12	3 unités de CPU
P2	T21	3 unités de CPU
	T22	2 unités de CPU
	T23	1 unité de CPU

Supposez que le processeur est libre. Donnez les diagrammes de Gantt montrant l'allocation du processeur, pour chacun des cas suivants :

a) Les threads sont supportés par le noyau (threads noyau). Le noyau ordonnance l'exécution des threads selon l'algorithme du tourniquet (Round Robin) avec un quantum de 2 unités.

La file d'attente des threads prêts, à l'instant t , est : $\rightarrow T23 \ T12 \ T22 \ T21 \ T11 \rightarrow$
T11 est en tête de file.

b) Les threads sont implémentés entièrement au niveau utilisateur. Le noyau ordonnance l'exécution des processus selon l'algorithme du tourniquet avec un quantum de 2 unités. Les threads sont ordonnancés au niveau utilisateur selon l'algorithme du tourniquet avec un quantum de 1 unité.

Supposez que le processeur est libre et que les processus sont passés à l'état prêt dans l'ordre suivant :

P1 puis P2

Dans P1 : T11 puis T12

Dans P2 : T21 puis T22 puis T23

Dans tous les cas, le temps de commutation est supposé nul.

c) Calculez, pour chaque cas, le temps de virement (temps de séjour), relatif à l'instant t , de chaque processus. Comparez puis commentez les résultats.

Exercice 5 :

Soit la charge de processus montrée sur la table ci-dessous. Pour chacun des algorithmes d'ordonnancement demandés :

- dessinez un diagramme montrant l'état d'occupation de l'UCT, des périphériques, et des files d'attente sachant que le système a un seul périphérique d'E/S commun à tous les processus.

- calculez les temps de séjour pour chaque processus ainsi que les temps de séjour moyen pour chaque cas d'étude.
- comparez les performances des divers algorithmes pour cette charge de processus.

Processus	Instant d'arrivée	Durée d'exécution CPU	Durée d'E/S	Instants d'E/S
P0	0	9	2	5 ET 8
P1	2	6	0	-
P2	5	5	3	6

- L'algorithme du premier-arrivé-premier-servi (PAPS).
- L'algorithme du tourniquet avec un quantum de 4 unités de temps et des niveaux de priorité externe de 3, 1, 2 respectivement pour P0, P1, P2.
- L'algorithme du tourniquet avec un quantum de 4 unités de temps et deux UCTs fonctionnant en parallèle.

Exercice 6:

Considérons un système d'exploitation doté d'un ordonnanceur préemptif, à priorité. Dans le système, il existe 20 niveaux de priorité, de 0 à 19, 0 étant la plus faible priorité. Supposons que le processeur exécute le jeu de processus présenté dans la table suivante :

Processus	Instant d'arrivée	Séquence d'exécution	Priorité
A	2	EEQE	10
B	5	EE	7
C	0	EQQQQE	3

Les processus A et C partagent la ressource Q, qu'ils accèdent en exclusion mutuelle.

Dans la séquence d'exécution d'un processus, Q signifie que pour une unité de temps le processus s'exécute et accède à la ressource Q, tandis que E signifie que pour une unité de temps le processus s'exécute sans accéder à la ressource Q.

- Définir brièvement la notion d'exclusion mutuelle.
- Donnez** le diagramme de Gantt pour l'ordonnancement des processus entre les instants 0 et 11. Sur le diagramme, indiquez les accès des processus à la ressource Q.
- Y a-t-il une inversion de priorité** sur le diagramme? Si oui, **précisez** l'instant de début, ainsi que la durée de l'inversion de priorité.