



Examen de Systèmes d'Exploitation 1

Date : 06/06/2022

Durée: 1h30 - Documentation non autorisée

Exercice 1: (Questions de Cours : 9 pts) (30 minutes)

Partie A) (QCM) Mettez une croix sur une seule réponse (1.5 pts).

1. l'ordonnement Rond Robin:

- Favorise les processus qui sont susceptibles d'avoir un temps CPU court.
- Favorise les processus à priorité élevée.
- Ajuste dynamiquement la priorité des processus en fonction de leur utilisation du CPU passé.
- Donne à chaque processus une part égale de temps CPU

2. Laquelle parmi les transitions suivantes n'est pas supportée par l'ordonnement sans préemption :

- Prêt → actif Actif → prêt Prêt → bloqué Bloqué → actif

3. Quand est-ce la préemption peut avoir lieu?

- Lorsqu'un quantum expire. Quand un processus demande une Entrée/Sortie.
- Lorsqu'un processus se termine sur erreur. Lorsque le processus moins prioritaire arrive.

Partie B) Qui suis-je ? (2.5 pts)

1. Je suis le premier programme qui est lancé à la mise sous tension de l'ordinateur.
2. Je suis une structure de donnée contenant toutes les informations relatives à un processus donné (PID, état, fichiers ouverts,...).
3. Je suis un appel système qui permet de créer un processus
4. Je suis un appel système qui affiche le PID du processus père.
5. Je suis la version avec réquisition de l'algorithme d'ordonnement SJF.

Partie C) (Questions de Compréhension : 5 pts)

1. Pourquoi l'algorithme d'ordonnement SJF n'est-il pas réellement applicable ?

.....

.....

.....

2. Quel est l'effet de la diminution du quantum sur les performances de l'algorithme RR (tourniquet)?

.....

.....

.....

.....

3. Qu'est-ce que la MMU ? Expliquer son rôle.

.....

4. Pour chacune des transitions suivantes entre les états possibles d'un processus, indiquez si la transition est possible (Mettez une X dans la bonne case).

	Transition possible.	Transition impossible.
(a) en cours d'exécution (actif) -> prêt		
(b) en cours d'exécution (actif) -> bloqué		
(c) prêt -> terminé		
(d) bloqué -> en cours d'exécution (actif)		
(e) bloqué -> terminé		

5. Sur un système utilisant la segmentation simple, calculez l'adresse physique de chacune des adresses logiques, à partir de la table des segments ci-après. On suppose que les adresses soient décimales au lieu de binaires.

Segment	Base	Limite
0	1100	500
1	2500	1000
2	200	600
3	4000	1200

Adresse logique	Adresse physique
<0, 300>
<2, 800>
<1, 600>
<1, 1111>

Exercice 2 : (Gestion des Processus : 8 pts) (40 minutes)

Partie A) (3 pts)

On considère une architecture monoprocesseur dans laquelle on désire exécuter l'ensemble des processus suivants :

Processus	A	B	C	D
Date d'arrivée	0	0	1	5
Durée	10	6	8	4

1. Donner les diagrammes de Gant et les temps de réponse moyen, en utilisant les algorithmes d'ordonnancement suivant :

- a) FCFS (first come first served),
- b) SJF (short job first).

FCFS	
TRM=	

SJF	
TRM=	

Partie B) (5 pts)

Considérons l'exécution de trois programmes A, B et C sur une configuration monoprocesseur (CPU, MC, périphériques d'E/S). On suppose qu'à l'instant $t=0$, la liste de processus à l'état prêt renferme les programme A, B et C. ceux-ci ont été soumis au système dans cet ordre et effectuant du calcul et des entrées/sortie selon les temps donnés ci-dessous :

	A	B	C
Temps d'exécution sur le CPU	3	4	3
E/S	7	2	3
Temps d'exécution sur le CPU	2	3	2
E/S	3	2	
Temps d'exécution sur le CPU	1	1	

- On considère que l'ordonnancement sur le processeur se fait selon une politique à priorité préemptible : le processus élu à un instant t est celui qui est le processus prêt de plus forte priorité. On donne : priorité (A) > priorité (B) > priorité (C). On considère que l'ordre de service des requêtes d'E/S pour le disque se fait toujours selon une politique FIFO (**Algorithme 1**).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CPU																									
File d'attente																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Disque																									
File d'attente																									

- On considère que l'ordonnancement sur le processeur se fait selon une politique tourniquet avec un quantum de 2 unités de temps. On suppose que l'ordre d'arrivée a été A puis B puis C. On considère que l'ordre de services des requêtes d'E/S pour le disque se fait en FIFO (**Algorithme 2**).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CPU																									
File d'attente																									
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Disque																									
File d'attente																									

3. Donnez les temps de réponse moyen pour chacun des 2 algorithmes précédents ? Calculer pour chaque algorithme le rendement (**R**) (ou le taux d'occupation) du processeur.

	TRM	R
Algorithme 1		
Algorithme 2		

Exercice 3 : (Gestion des processus : 3 pts) (20 minutes)

<p>1. Quel est le résultat d'exécution de ce code (1.5 pts)</p>	<p>2. Préciser le nombre de processus créer et dessiner l'arbre généalogique des processus engendrés par le programme suivant : (1.5 pts)</p>
<pre>int main (void) { pid_t p=1 ; while (p>0) p=fork(); printf (" Je suis %d \n " , getpid()); return 0 ; }</pre>	<pre>int main() { int temp = 0; while (temp < 5) { if (fork() > 0) temp++; else temp = 5; }</pre>
<p style="text-align: center;"><u>Résultat d'exécution</u></p>	<p style="text-align: center;"><u>L'arbre généalogique</u></p> <p style="text-align: right;">Nombre de processus créer :</p>



Bon courage