

Année d'étude: 2eme Année Licence Informatique

Examen de systèmes d'exploitation 1

Exercice 1: (Questions de cours : 5 pts)

(Les exercices (1) et (2) sont à comptabiliser pour l'interrogation pour les étudiants ayants ratés l'interrogation).

Q1) Choisir la bonne réponse. (Barème : - Une bonne réponse = +1, - Pas de réponse = 0, - Une mauvaise réponse = - 0.5)

La différence entre l'ordonnancement de programmes avec préemption et sans préemption est :

- (a) Si ou non un processus prêt peut être involontairement terminé depuis l'état prêt.
- (b) Si ou non un processus élu est involontairement enlevé de l'état actif.
- (c) Si ou non un processus bloqué peut avoir ses ressources involontairement lui retirées.

Q2) Décrire (sur 2 à 3 phrases) la différence entre les algorithmes d'ordonnancement de programmes préemptifs et non préemptifs. Lequel qui convient mieux pour un système à temps partagé?

Q3) On s'intéresse à classer divers services système selon leur appartenance. Pour cela on vous demande de donner votre réponse sous forme d'un tableau rempli comme le suivant, en mettant une croix dans la case qui convient.

Service système	Noyau système	Utilitaire système
1) Traduire un programme écrit en Visuel Basic		
2) Renommer un répertoire		
3) Compresser les données d'un fichier		
4) Lire la position courante de la souris		

Q4) Supposons qu'on a une machine avec 4 processeurs et 3 processus qui sont en état « prêt ». Quelle est d'après vous la politique d'ordonnancement de processus qui donnerait les meilleures performances pour le système ? Argumentez votre réponse.

Exercice 2 : (6 pts)

Six étudiants du département Informatique viennent voir l'enseignant X (nous tenons à garder son anonymat). L'enseignant X dispose d'une heure pour discuter avec les étudiants et passer d'un étudiant à un autre lui prend **une minute**. La table suivante donne l'ordre d'arrivée des étudiants et le temps dont ils ont besoin. Les étudiants arrivent tous au même moment.

Etudiant	Ordre d'arrivée	Temps requis
e1	1	45min
e2	2	15 min
e3	3	10 min
e4	4	5 min
e5	5	6 min
e6	6	20min

- 1) Sachant que l'enseignant X ne peut consacrer en tout qu'**une heure** à l'ensemble des étudiants, quel est l'algorithme d'ordonnement qui lui permettra de traiter **le plus d'étudiants complètement**. Justifier à l'aide des diagrammes.
 - a) premier arrivé, premier servi (*FCFS*)
 - b) le plus court d'abord (*SJF*)
 - c) Tourniquet ou *Round Robin* avec un quantum=5mn. (l'intervalle de **1mn** reste valable même si l'enseignant X reste avec le même étudiant pendant 2 quantums de suite)

- 2) Un ordonnancement de type tourniquet peut utiliser différents quantums. Donner une raison d'utiliser un petit quantum ainsi qu'une raison d'utiliser un grand quantum.

Exercice 3: (4 pts)

Supposons deux processus P1 et P2 entrent dans la file d'attente des processus prêts avec les propriétés suivantes : Le processus P1 a un total de 8 unités de temps CPU à exécuter, mais après chaque 2 unités CPU il doit exécuter 1 unité d'entrée/sortie (*ainsi la durée minimal d'accomplissement de ce processus est 12 unités*). Le processus 2 a un total de 20 unités CPU à exécuter. Ce processus arrive juste derrière le processus P1. Donner les diagrammes de GANTT résultants en utilisant les algorithmes d'ordonnements suivants:

- 1) SRTF (appelé aussi SJF avec préemption)
- 2) Round-Robin avec un quantum de 4 unités CPU.

Exercice4 : (5 pts)

Supposez que le système d'exploitation est composé de deux unités de contrôle (deux processeurs CPU1 et CPU2) et d'une unité d'E/S. Tous les processus prêts sont dans une même file d'attente. La commutation de contexte est supposée de durée nulle.

Considérez trois processus A, B et C décrits dans le tableau suivant :

processus	Instant d'arrivée	Temps d'exécution
A	0	4 unités cpu, 2 unités d'E/S, 2 unités cpu
B	2	3 unités cpu, 4 unités d'E/S, 2 unités cpu
C	3.5	5 unités cpu

La première ligne signifie que le processus A arrive dans le système à l'instant 0, son exécution nécessite dans l'ordre 4 unités de temps CPU, 2 unités de temps d'E/S et 2 unités de temps CPU.

Au départ le processus A est élu par le processeur CPU1.

Si plusieurs événements surviennent en même temps, vous supposerez les priorités suivantes :

- Le CPU1 a la priorité d'accès à la file des processus prêts par rapport au CPU2.
- A la fin d'un quantum, le processus en cours et non terminé est suspendu uniquement si la file des processus prêts n'est pas vide. Le traitement réalisé à la fin d'un quantum est plus prioritaire que celui d'une fin d'E/S qui, à son tour, est plus prioritaire que l'arrivée de nouveaux processus dans le système.

1) Donnez les diagrammes d'exécution des jobs A, B et C montrant l'allocation des deux processeurs, de l'unité d'E/S dans le cas des systèmes : multiprogrammé et temps partagé avec un quantum de temps égal à 3 unités CPU.

2) Calculez le temps moyen de réponse pour chaque système.

Bon courage

Corrigé de l'Examen de systèmes d'exploitation 1

Exercice n° 1: (Questions de cours : 5 pts)

Q1) Réponse : (b)

Q2) Réponse :

- L'ordonnancement de programme préemptif permet à un processus d'être interrompu au milieu de son exécution, en lui retirant la cpu et l'allouant à un autre processus. (0.25)
- L'ordonnancement non préemptif s'assure qu'un processus lâche l'unité centrale de traitement seulement quand il termine avec son exécution en cours. (0.25).
- C'est les algorithmes préemptifs (ou bien round robin) qui conviennent au système à temps partagé. (0.5pt)

Q3)

Service système

Noyau système Utilitaire système

3. Traduire un programme écrit en Visuel Basic

x

4. Renommer un répertoire

x

5. Compresser les données d'un fichier

x

6. Lire la position courante de la souris

x

Q4)

Réponse : Etant donné qu'il y a plus de processeurs que de processus, n'importe quelle stratégie d'ordonnancement qui affecte entièrement un processeur à un processus donne de bons résultats. Il y a plus de ressources que de demandes.

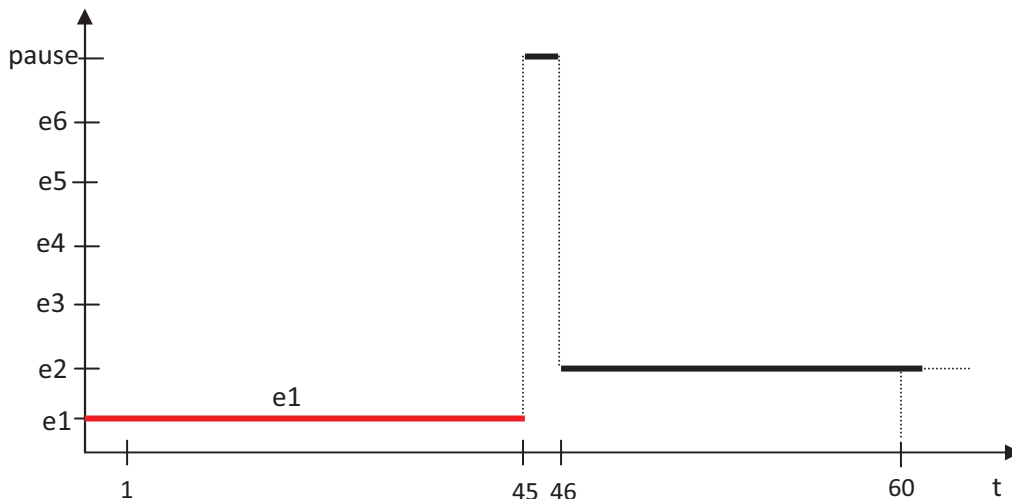
Exercice 2

1) Réponse : le plus court d'abord (0.5pts)

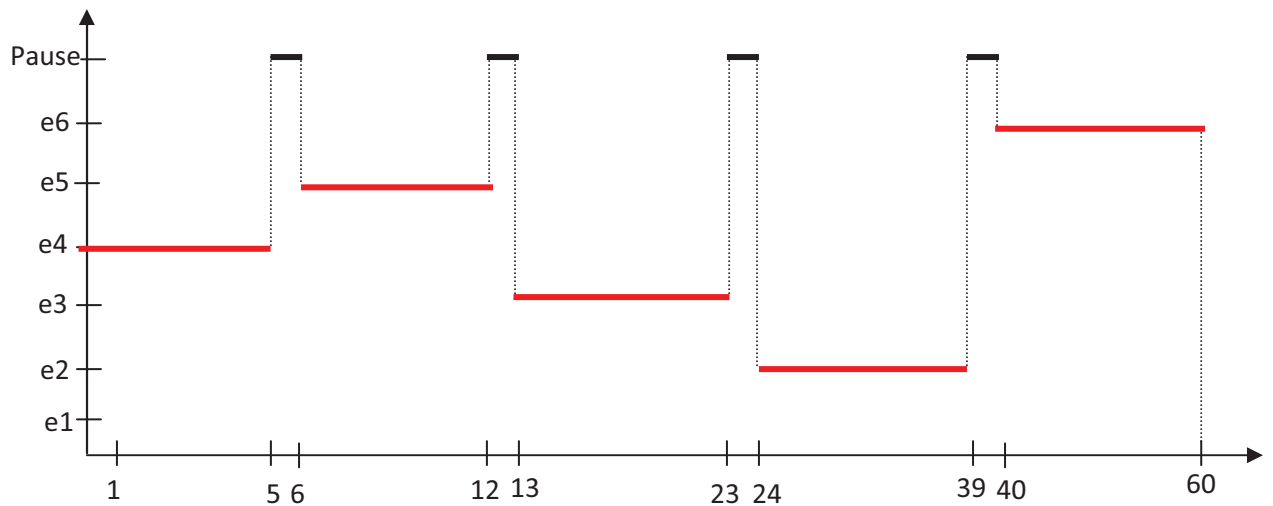
Justification :

a) Premier arrivé, premier servi : 1 étudiant complètement traité (0.25*3)

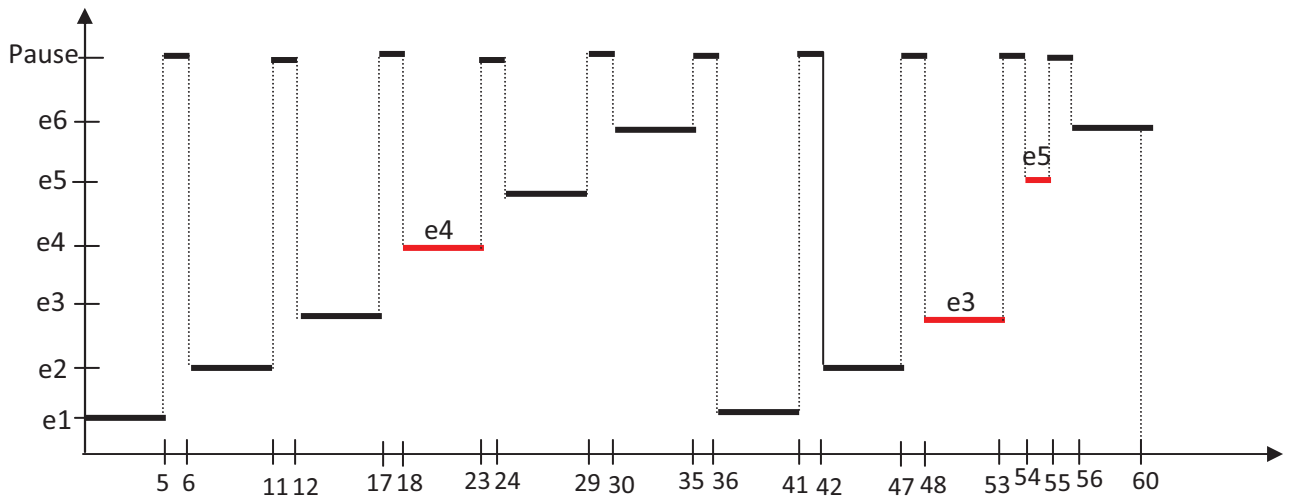
(Barème : Chaque segment correct est noté sur 0.25 pts)



b) Le plus court d'abord : 5 étudiants complètement traités (0.25*9)



c) Tourniquet (quantum=5mn) : 3 étudiants complètement traité (0.25*10)

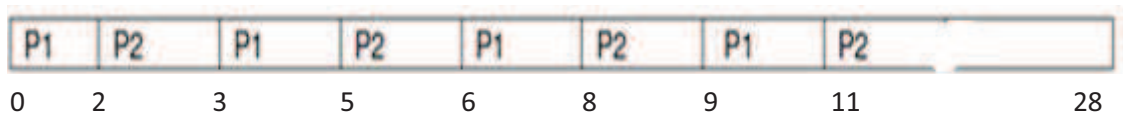


1) *Un petit quantum assure un temps de réponse rapide (plus d'interactivité). un grand quantum minimise le temps de changement de contexte (augmente le rendement du pc).*
(0.25*2)

Exercice 3: (4 pts)

Barème : chaque segment correct est noté sur 0.25 pts

1) **SRTF** (0.25*8=2pts)



Entre 2-3 unité de temps, p2 est dans la file des processus prêt et P1 est allé faire E/S. ainsi P2 devrait être exécuté comme c'est un algorithme SJF préemptif.

2) RR(0.25*8=2pts)

P1	P2	P1	P2	P1	P2	P1	P2		
0	2	6	8	12	14	18	20		28

EXERCICE 4 : (4.5 pts)

Barème : chaque segment correct est noté sur 0.25

1)

a) Multiprogrammé :

E/S					A	A	B	B	B	B			
Cpu2			B	B	B		A	A					
Cpu1	A	A	A	A	C	C	C	C	C		B	B	
Temps	0	1	2	3	4	5	6	7	8	9	10	11	12

Ou bien :

Axe cpu1 : 0-4 :A, 5-9 : C, 10-12 : B

Axe cpu2 : 2-5 :B, 6-8 :A

Axe E/S : 4-6 :A, 6-10 :B

(Barème : axe cpu1 : 0.25*3 ; axe cpu2 : 0.25*2 ; axe E/S : 0.25*2 : total 1.75 pt)

b) Temps partagé avec une Tranches de temps =3 unités cpu

E/S					A	A	B	B	B	B			
Cpu2			B	B	B		A	A					
Cpu1	A	A	A	A	C	C	C	C	C		B	B	
Temps	0	1	2	3	4	5	6	7	8	9	10	11	12

2)

a) Système multiprogrammé :

Temps de réponse= temps de fin exécution –temps d’arrivé

Temps de réponse moyen= la somme des temps de réponse de chaque job/ le nombre de jobs.

= (8-0) + (12-2) + (9-3.5)/3= 8+10+6.5/3 (0.25 pour la formule de calcule)

=23.5/3=7.8. (0.25 pour le résultat correct obtenu)

b) Système à temps partagé : même results trouvé en reponse 1 (0.5pt)