

Exercice 1

Soient trois (03) processus P1, P2 et P3 décrits comme suit :

	Temps d'exécution (ms)	Date d'arrivée (ms)
P1	3 (2) 1 [2] 1 [4]	0
P2	[1] 3. (3)	0
P3	3 (2) 1	0

NB :

- x (y) z signifie que le processus fait x ms calcul, ensuite y ms E/S sur le Périph1 et z ms calcul.
- x [y] z signifie que le processus fait x ms calcul, ensuite y ms E/S sur le Périph2 et z ms calcul.

Sachant que :

- Le système dispose de deux périphériques d'E/S, Périph1 et Périph2.
- Le Périph1 est plus prioritaire que le Périph2.
- Une tâche de contrôle système s'effectue en 1ms quelque soit sa nature.
- Le quantum de temps est égal à 3 ms.

1. Donner la synoptique d'exécution dans le mode d'exploitation multiprogrammé et temps partagé.
2. Donner le temps de réponse moyen et le temps d'attente moyen.

Exercice 2

Considérons la politique de scheduling du système Linux doté d'un scheduler préemptif, à base de priorité. Dans le système, il existe 100 niveaux de priorité, de 0 à 99, 0 étant la plus faible priorité.

Sous ce système, les processus sont répartis en trois classes :

- **SCHED_FIFO** : dans cette classe, le processeur est donné au processus de plus haute priorité. Ce processus peut être préempté par un processus de la même classe ayant une priorité supérieure.
- **SCHED_RR** : dans cette classe, le processeur est donné au processus de plus haute priorité pour un **quantum** de temps égal à 20 ms. La politique appliquée est celle du tourniquet.

SCHED_OTHER : cette classe accueille les processus ayant une priorité nulle. Le processeur est donné au processus se trouvant en tête de file pour un **quantum** de temps égal à 10 ms. La politique appliquée est celle du tourniquet. Tout processus exprimant une demande d'E/S, passe à la classe **SCHED_RR**

avec une nouvelle priorité calculée par la formule $Pri(P_i) = Pri(P_i) + 20$.

Les processus de la classe **SCHED_FIFO** sont plus prioritaires que les processus de la classe **SCHED_RR** et qui eux-mêmes sont plus prioritaires que ceux de la classe **SCHED_OTHER**.

Soient six processus P1, ..., P6 décrits comme suit :

Processus	Classe	Instant d'arrivée (ms)	Durée d'exécution (ms)	Priorité
P1	SCHED_RR	0	30(40)30	18
P2	SCHED_FIFO	0	20(30)10	85
P3	SCHED_RR	30	20	18
P4	SCHED_OTHER	55	10(20)30	0
P5	SCHED_OTHER	55	30	0
P6	SCHED_FIFO	150	20(10)10	58

1. Donner le diagramme d'exécution pour l'ordonnancement de ces processus en appliquant la politique décrite ci-dessus.
2. Donner l'état des différents processus aux instants : 30 ms, 55 ms, 120 ms, 135 ms et 160 ms.
3. Donner l'état des files d'attente **SCHED_FIFO**, **SCHED_RR** et **SCHED_OTHER** aux instants : 30 ms, 55 ms, 120 ms, 135 ms et 160 ms.
4. Ecrire le pseudo code du **Scheduler** et le **SVC**.
5. Ecrire le pseudo code du **pilote** dans le cas d'une **sortie asynchrone** de N caractères.

Exercice 3

On considère un système qui utilise un mécanisme de pagination à la demande pour la gestion de la mémoire. Soit un programme possédant un espace virtuel de 512 octets et qui fait référence, durant son exécution, aux adresses virtuelles suivantes :

34 205 123 510 145 456 345 412 10 258 12 234 336 412

1. Sachant que la taille de la mémoire physique est de 192 octets et la taille de la page est de 64 octets, calculer le taux de défauts de pages générés en appliquant les algorithmes de remplacement **FIFO**, **OPTIMAL** et **LRU**.