

★ Exercice 1:

5 travaux A, B, C, D et E sont soumis à un calculateur dans cet ordre, mais quasi simultanément. Ces travaux ne font pas d'entrées-sorties. Leurs durées respectives sont 10, 6, 2, 4 et 8 secondes.

▷ **Question 1:** Déterminer les temps de réponse de chacun des travaux, ainsi que le temps de réponse moyen, pour les disciplines FIFO (First In First Out) et SJF (Shortest Job First). Donner aussi le délai de rotation et le temps d'attente.

▷ **Question 2:** Même question pour une discipline à priorité (sans réquisition), avec $P(A) = 3$, $P(B) = 5$, $P(C) = 2$, $P(D) = 1$, $P(E) = 4$. Une valeur basse correspond à une priorité plus élevée.

▷ **Question 3:** Même question avec la discipline PS (Process Sharing - Tourniquet) et un quantum de 2 secondes (sans priorités)

★ Exercice 2:

Pour les processus du tableau ci-dessous, dessinez un schéma illustrant leur exécution en utilisant l'ordonnancement de priorité dans les 2 cas :

- non préemptif
- préemptif

Remarque pour cette question, un nombre de priorité élevé correspond à une priorité plus importante. La date d'arrivée est exprimée en cycles.

Processus	Date d'arrivée	Cycles	Priorité
A	0	4	3
B	1	3	4
C	2	3	6
D	3	5	5

★ **Exercice 3:** Sous Linux 2.6, soit un processus P qui fait des calculs pendant 60 ms crée 5 processus A, B, C, D et E pratiquement en même temps. Ensuite, il attend la fin de tous ses fils avant de faire 30 ms de calcul et terminer.

Il met A, C et E dans la classe SCHED_RR avec les priorités 4, 2 et 2 respectivement. Il met B et D dans la classe SCHED_FIFO avec les priorités 3 et 1 respectivement.

On suppose qu'il n'y a que ces processus dans le système. Si 2 processus ont la même priorité, on fera l'ordonnancement par ordre alphabétique.

Donner le schéma d'ordonnancement correspondant sachant que :

- A : calcul(100)
- B : calcul(90), usleep(30), calcul(10)
- C : calcul(40), fait sched_setscheduler() pour passer en SCHED_NORMAL, calcul(30)
- D : calcul(70)
- E : calcul(30)

Le quantum de la classe SCHED_RR est 10. L'unité de temps utilisée est le milliseconde. Le schéma d'ordonnancement est à donner selon le format suivant :

$X(T_X)-Y(T_Y)-\dots$

X est le nom du processus et T_X est le temps alloué à ce processus.

★ **Exercice 4:** On considère le système d'exploitation Linux version 2.6. Soient 4 processus A, B, C et D créés pratiquement en même temps avec les valeurs de nice 0, -4, -16 et 4 respectivement. Tous appartiennent à la classe SCHED_NORMAL et ont le comportement suivant (boucle infinie) :

- **A** : 100 ms calcul, 800 ms E/S
- **B** : 400 ms calcul, 50 ms E/S
- **C** : que des calculs, pas d'E/S
- **D** : que des calculs, pas d'E/S

▷ **Question 1:** Remplir les colonnes 2, 3, 4 et 6 du tableau suivant :

Processus	nice	SP (static priority)	quantum (timeslice)	DP_0 (init. dynamic priority)	DP_1	DP_2
A						
B						
C						
D						

▷ **Question 2:** Sur quoi le scheduler Linux 2.6 se base-t-il pour choisir le prochain processus à exécuter ?

▷ **Question 3:** Quand est-ce qu'un processus est considéré comme interactif ? Quel traitement particulier est réservé à un tel processus ? Quels processus parmi A, B, C et D ne pourront jamais être considérés comme interactifs ? Justifier ! Remplir la colonne 5 du tableau précédent.

▷ **Question 4:** Donner le schéma d'ordonnancement correspondant en remplissant le tableau suivant. Dans la file d'attente active, on respecte l'ordre des priorités comme déjà fait dans la première ligne du tableau. Pour simplifier, l'ordonnancement ici est supposé non préemptif. Renseigner les colonnes relatives à la priorité dynamique du tableau précédent au fur et à mesure.

[illegible]