

## Chapitre 9 : Ordonnancement

- 9.1 Cinq tâches de longueur 9, 6, 3, 5 et 7 doivent s'exécuter. Quel ordonnancement de ces tâches produira le plus petit temps d'attente moyen?

Afin de minimiser le temps d'attente moyen, il suffit d'exécuter les tâches les plus courtes en premier: 3, 5, 6, 7, 9.

- 9.2 Cinq tâches P1 à P5 ont les temps d'exécution suivants avec la priorité fournie entre parenthèses : 10(3), 6(5), 2(2), 4(1), 8(4). Quel sera le temps moyen de service selon l'algorithme d'ordonnancement : tourniquet, priorité, premier arrivé premier servi, le plus court en premier?

Avec le tourniquet, chaque tâche obtient le  $\frac{1}{5}$  du temps. Après 10 unités, la tâche de durée 2 a terminé et le temps se divise en  $\frac{1}{4}$  pour 8 unités. Par la suite, le temps se divise en  $\frac{1}{3}$  pour 6 unités, puis en  $\frac{1}{2}$  pour 4 unités et la dernière finit en 2 unités ensuite. La séquence donne 0 :P1, 1 :P1, 2 :P2, 3 :P2, 4 :P3, 5 :P3(fin), 6 :P4, 7 :P4, 8 :P5, 9 :P5, 10 :P1, 11 :P1, 12:P2, 13:P2, 14:P4, 15:P4(fin), 16:P5, 17:P5, 18:P1, 19:P1, 20:P2, 21:P2(fin), 22:P5, 23:P5, 24:P1, 25:P1, 26:P5, 27:P5(fin), 28:P1, 29:P1(fin). Les temps de fin sont donc 6, 16, 22, 28, 30 pour une moyenne de 20.4. En fonction de la priorité, les tâches finissent dans l'ordre de priorité, au temps 6, 14, 24, 26, 30, pour une moyenne de 20. Si les tâches sont servies en ordre, on obtient 10, 16, 18, 22, 30, pour une moyenne de 19.2. La tâche la plus courte en premier donne 2, 6, 12, 20, 30 pour une moyenne de 15.

- 9.3 La prédiction avec pondération décroissante selon l'âge est utilisée avec un facteur  $\alpha=0.5$ . Les temps précédents furent 40, 20, 40, 15ms. Quelle sera la prédiction pour le prochain temps?

Les prédictions seront de  $40, 40 \cdot 0.5 + 0.5 \cdot 20 = 30, 0.5 \cdot 0.5 \cdot 40 + 0.5 \cdot 0.5 \cdot 20 + 0.5 \cdot 40 = 35$  (ou  $0.5 \cdot 30 + 0.5 \cdot 40 = 35$ ),  $0.5 \cdot 0.5 \cdot 0.5 \cdot 40 + 0.5 \cdot 0.5 \cdot 0.5 \cdot 20 + 0.5 \cdot 0.5 \cdot 40 + 0.5 \cdot 15 = 25$ .

- 9.4 Un système en temps réel comporte 4 tâches périodiques de période/durée : 50/35, 100/20, 200/10 et 250/x ms. Quelle est la plus grande valeur de x pour laquelle le système peut être ordonnancé?

Les taux d'utilisation du CPU sont  $35/50 + 20/100 + 10/200 = .95$ . Il ne reste donc que 5% du temps disponible, soit 12.5ms/250ms.

- 9.5 Pourquoi un système d'ordonnancement à deux niveaux est-il souvent utilisé?

Sur les systèmes avec de grandes queues de tâches à exécuter (serveur de calcul) ou avec peu de mémoire, il peut être intéressant d'avoir un premier niveau d'ordonnancement pour admettre une tâche en mémoire. Même une fois admis en mémoire, les calculs d'ordonnancement peuvent se faire à deux

niveaux, au niveau le plus bas simplement prendre la prochaine tâche de plus haute priorité, à un niveau plus haut, ajuster les priorités ou migrer les tâches d'un CPU à l'autre pour équilibrer la charge.

9.6 Comment peut-on permettre de changer facilement la politique d'ordonnement, tout en conservant l'essentiel du mécanisme intact?

Il est relativement facile d'avoir un mécanisme de bas niveau qui exécute simplement pour la période en cours les tâches selon leur priorité, alors qu'un module de plus haut niveau, possiblement paramétrisable ou interchangeable, se charge d'ajuster les priorités selon le comportement du système et les objectifs et politiques d'ordonnement souhaités.