

TD Ordonnancement des Processus

Exercice 1. (Exam rat. 06/2013)

On considère trois (3) processus P1, P2, P3 dont les durées d'exécution sont respectivement 6, 4 et 8 unités de temps.

On fait les hypothèses suivantes :

H1 : Après 1 unité de temps d'exécution, le processus P2 crée un processus fils (qu'on appellera P4) dont la durée d'exécution est de 3 unités de temps.

H2 : Le processus P4 après 2 unités de temps d'exécution crée à son tour un nouveau processus fils P5, dont la durée d'exécution est de 2 unités de temps.

H3 : Un processus ayant créé un fils doit se bloquer jusqu'à la terminaison de son processus fils.

- 1) En supposant que tous les processus sont gérés en utilisant le scheduling «Round-Robin» avec un quantum égal à 2 unités de temps :
 - a) Dessinez le diagramme de Gantt.
 - b) En déduire les temps d'arrivée des processus P4 et P5 ainsi que les temps de réponse de chaque processus.
- 2) On relâche maintenant l'hypothèse H3 et on considère qu'un processus ayant créé un fils continue de s'exécuter mais à sa fin il doit se bloquer en attente de son fils. Reprendre la question 1 pour cette nouvelle hypothèse.

Exercice 2. (Exam. Rat. 6/2011)

Considérons les 6 processus suivants, à être exécutés sur un monoprocesseur :

Processus	Temps d'arrivée	Temps de traitement (ms)	Priorité
P1	0	10	2 (Argent)
P2	2	8	1 (Or)
P3	3	3	3 (Bronze)
P4	10	4	2 (Argent)
P5	12	1	3 (Bronze)
P6	15	4	1 (Or)

- 1) Donnez les diagrammes de Gantt montrant l'exécution de ces différents processus en utilisant les algorithmes d'ordonnancement préemptifs suivants : priorité; et SRTF

- 2) Lequel des algorithmes d'ordonnement de la question 1) a fourni le temps de vie moyen le plus bas? En déduire, pour chaque processus, le temps d'attente.

Exercice 2. (exam. 01/2011)

Dans cet exercice on suppose un ordonnanceur qui admet les caractéristiques suivantes :

- Plus la valeur de priorité est grande et plus la priorité est grande.
- L'ordonnement est basé sur une priorité préemptive.

Soient trois processus P1, P2, et P3, ayant comme les caractéristiques et les codes suivants :

P1	P2	P3
(Priorité =1, prêt à t=0)	(Priorité =2, prêt à t=4)	(Priorité =3, prêt à t=3)
1. S'exécute pendant 2 u	1. S'exécute pendant 1 u	1. S'exécute pendant 2 u
2. Fait P(S1)	2. Fait V(S2)	2. Fait P(S2)
3. S'exécute pendant 2 u	3. S'exécute pendant 1 u	3. S'exécute pendant 2 u
4. Fait V(S1)		4. Fait P(S1)
5. S'exécute pendant 1 u		5. S'exécute pendant 1 u

Où u est égale à 1 unité de temps, S1, S2 sont des sémaphores dont les valeurs initiales sont respectivement 1 et 0, et les opérations P et V s'exécutent pendant un temps nul.

- 1) Donnez sous la forme d'un tableau commenté la séquence d'exécution de ces trois processus. A t=0, c'est P1 qui démarre son exécution.
- 2) Donnez le diagramme de GANTT correspondant à cet ordonnancement.

Exercice 3. (DS 11/2010)

On considère trois (3) processus P1, P2, P3 dont les durées d'exécution sont respectivement 6, 4 et 8 unités de temps.

On fait l'hypothèse suivante : après 1 unité de temps d'exécution, le processus P2 crée, par *fork()*, un processus fils (qu'on appellera P4) dont la durée d'exécution est de 3 unités de temps. Le processus P4 après 2 unités de temps d'exécution crée à son tour un nouveau processus fils P5, dont la durée d'exécution est de 2 unités de temps. On admet qu'un processus ayant créé un fils doit se bloquer jusqu'à la terminaison de son processus fils.

En supposant que tous les processus sont gérés en utilisant la politique d'ordonnement «Round-Robin» avec un quantum égal à 2 unités de temps et un **temps de commutation qui est égal à 1 unité de temps**.

- 1) Dessinez le digramme de Gantt.
- 2) En déduire pour chaque processus : le temps d'attente et le temps de vie.

Exercice 4. (exam. 01/2010)

L'ordonnement EDF (Earliest Deadline First) est un algorithme d'ordonnement temps réel de processus. A chaque fois qu'un processus demande du temps CPU, il doit préciser une échéance (une date limite >0). Le processus doit avoir obtenu son temps CPU avant d'atteindre la date limite.

L'ordonnanceur vise à satisfaire les demandes avant leurs échéances. Pour se faire, il gère une liste des processus prêts, classés par ordre croissant des échéances. L'algorithme exécute le premier processus de la liste qui correspond à celui dont l'échéance est la plus proche.

Si deux processus ont la même échéance, l'ordonnanceur sélectionne le plus prioritaire.

Lorsqu'un processus devient prêt ou arrive dans le système, le système vérifie si son échéance est antérieure ou égale à celle du processus en cours d'exécution. Le processus en cours est préempté pour l'une des deux raisons suivantes :

- L'échéance du processus en cours est plus loin que celle d'un processus prêt.
- L'échéance du processus en cours est la même que celle d'un processus prêt plus prioritaire.

Considérez trois processus A, B et C suivants :

- A demande le CPU toutes les 20ms et requiert à chaque fois 4 ms de temps CPU.
- B demande du temps CPU toutes les 30ms et requiert à chaque fois 10 ms.
- C demande du temps CPU toutes les 40ms et a besoin à chaque fois 20 ms de temps CPU.
- A est supposé plus prioritaire que B qui est, à son tour, plus prioritaire que C.

Supposons qu'au départ, les processus sont prêts et que l'échéance de chaque demande est la date de la prochaine demande. Par exemple, pour A, les dates d'échéances sont successivement 20, 2*20, 3*20,

Donnez le diagramme de Gantt pour les 90 premières ms, en précisant à chaque fois le contenu de la file d'attente des processus prêts.

Exercice 5. Examen 01 / 2008

On considère un système monoprocesseur et les 4 processus P1, P2, P3 et P4 qui effectuent du calcul et des E/S (disque) selon les temps donnés ci-dessous :

Processus P1

Calcul : 3 unités de temps

E/S : 7 unités de temps

Calcul : 2 unités de temps

E/S : 1 unité de temps

Calcul : 1 unité de temps

Processus P3

Calcul : 2 unités de temps

E/S : 3 unités de temps

Calcul : 2 unités de temps

Processus P2

Calcul : 4 unités de temps

E/S : 2 unités de temps

Calcul : 3 unités de temps

E/S : 1 unité de temps

Calcul : 1 unité de temps

Processus P4

Calcul : 7 unités de temps

On considère que l'ordonnancement sur le processeur se fait selon une *politique à priorité préemptible*: le processus élu à un instant t est le processus prêt de plus forte priorité. On donne priorité (P1) > priorité (P3) > priorité (P2) > priorité (P4).

On suppose que l'ordre de services des requêtes d'E/S pour le disque se fait également selon la *priorité des processus*: le processus commençant une E/S est celui de plus forte priorité parmi ceux en état d'attente du disque. Une opération d'E/S commencée ne peut pas être préemptée.

Sur le graphique 1 (page 5), donnez le chronogramme d'exécution des 4 processus P1, P2, P3 et P4. Vous distinguerez les états des processus : Prêt, Actif et Bloqué et vous indiquerez le contenu des files d'attente des processus (attente processeur et attente du disque). Pour vous guider, la première unité de temps est déjà portée sur le chronogramme. Justifiez votre raisonnement, en expliquant les transitions des processus. Donnez le temps de vie moyen obtenu.

Graphique 1

