

T-NSI OS et Processus

EXERCICE #1 Problème d'accès concurrent

Deux agences d'une banque veulent mettre à jour le même compte bancaire. Pour cela, l'agence de Nancy effectue :

1. courant = get_account(1867A)
 2. nouveau = courant + 1000
 3. update_account(1867A, nouveau)
- et l'agence de Karlsruhe :
- A. aktuelles = get_account(1867A)
 - B. neue = aktuelles -1000
 - C. update_account(1867A, neue)

En supposant que l'agence de Nancy commence en premier, quel sera le montant à l'issue des transactions?

Ordonnement

EXERCICE #2 Préemptif avec priorités

Nous utilisons un ordonnancement préemptif avec priorité

Nous allons utiliser un jeu de tâches qui mélange des tâches périodiques et des tâches ponctuelles.

Tâche	Date(s) d'arrivée(s)	Priorité	Durée	Remarque
A	0, 6, 12, 18, 24, 30 ...	10	1	Périodique
B	0, 10, 20, 30 ...	8	4	Périodique
C	21	9	6	Ponctuelle
D	0	1	7	Ponctuelle

1. Faire l'ordonnement de ces tâches sur 32 unités de temps.
2. Quel est le temps de réponse de chaque tâche ?

EXERCICE # 3 Ordonnement collaboratif

Le système n'interrompt jamais une tâche, il ne reprend la main que si la tâche se termine ou fait une opération bloquante. Supposons le jeu de tâches suivant :

Tâche	Date(s) d'arrivée(s)	Priorité	Durée	Remarque
A	0	4	5	au bout de 3 unités de calculs, la tâche laisse la main (yield) elle est immédiatement prête
B	1	8	4	au bout de 2 unités de calculs, la tâche fait un read bloquant qui dure au moins 3 UC
C	10	9	3	
D	1	5	4	

Faire l'ordonnement sur 16 unités de temps.

EXERCICE #4 Ordonnement avec liste de priorité

- Le système gère des listes de priorités.
- On se place en mode préemptif c'est à dire que lorsqu'une tâche plus prioritaire arrive, le système lui donne immédiatement accès au processeur.
- Pour des tâches de même priorité, le système utilise le Round Robin avec un quantum de 2.

Tâche	Date(s) d'arrivée(s)	Priorité	Durée	Remarque
A	0	4	5	
B	1	4	4	
C	3	8	6	

Faire l'ordonnement sur 15 unités de temps.