

Ordonnancement de tâches

Le surcoût de préemption et de calcul lié à l'algorithme d'ordonnancement utilisé est supposé négligeable pour les exercices de ce TD, il sera considéré nul.

Des chronogrammes vierges sont disponibles en annexe, une case correspond à un quantum de temps. Utilisez-les en noircissant les cases où les tâches sont dans l'état d'exécution.

Exercice 1 Etats et transitions d'une tâche

Déterminez si chacune des transitions suivantes entre états d'une tâche est possible (précisez un élément qui pourrait en être à l'origine le cas échéant) :

- Q 1.1 Exécuté – Prêt
- Q 1.2 Bloqué – Prêt
- Q 1.3 Bloqué – Exécuté
- Q 1.4 Exécuté – Inexistant

Exercice 2 Ordonnements de tâches aperiodiques

Considérons un système monoprocesseur exécutant cinq tâches indépendantes $\tau_1, \tau_2, \tau_3, \tau_4$ et τ_5 dont les propriétés sont les suivantes :

Tâche	Date d'activation (r)	Durée d'exécution (C)	Priorité
τ_1	T_0	$5q$	3
τ_2	T_0	$3q$	5
τ_3	T_0	q	2
τ_4	T_0	$2q$	1
τ_5	T_0	$4q$	4

Un quantum de temps q vaut 2ms.

Déterminez la séquence d'ordonnement (le chronogramme), le temps de réponse de chacune des tâches ainsi que le temps de réponse moyen pour les cas suivants :

- Q 2.1 Un SE avec ordonnancement *First In First Out* (FIFO) ;
- Q 2.2 Un SE avec ordonnancement *Shortest Job First* (SJF) ;
- Q 2.3 Un SE avec ordonnancement à priorité fixe et sans préemption ;
- Q 2.4 Un SE avec ordonnancement *Round Robin* (RR).

Notez qu'une valeur élevée de priorité indique un intérêt faible, ainsi la valeur la plus petite correspond à la tâche la plus prioritaire. Si nécessaire (cela dépend de l'algorithme utilisé), l'ordre d'activation des tâches est $\tau_1, \tau_2, \tau_3, \tau_4$ puis τ_5 .

Exercice 3 Ordonnancements de tâches périodiques

Toujours dans le cadre d'un système monoprocesseur, considérons désormais les trois tâches indépendantes, périodiques et à échéances sur requêtes τ_6 , τ_7 et τ_8 :

Tâche	Date d'activation (r)	Période ($P = R$)	Durée d'exécution (C)
τ_6	T_0	$6q$	$2q$
τ_7	T_0	$8q$	$2q$
τ_8	T_0	$12q$	$3q$

Un quantum de temps q vaut maintenant 1ms.

Considérons les deux types d'ordonnement préemptifs suivants :

- *Rate Monotonic* (RM) à priorité fixe selon la période d'activation des tâches ;
- *Least Laxity First* (LLF) à priorité dynamique selon le relâchement des tâches.

Dans l'algorithme d'ordonnement RM, la priorité maximale à l'instant t est accordée à la tâche qui a la plus petite période.

Une condition suffisante pour l'ordonnabilité d'un système avec RM est :

$$U = \sum_{i=1}^n \frac{C_i}{P_i} \leq n(2^{\frac{1}{n}} - 1)$$

- Q 3.1 Pouvez-vous être certain a priori, c'est-à-dire sans calculer la séquence d'ordonnement, que le système étudié est ordonnable avec l'algorithme RM ?
- Q 3.2 Déterminez la séquence d'ordonnement du système avec l'algorithme RM.

Dans le cas de l'algorithme d'ordonnement LLF, à chaque événement de tâches (arrivée, activation, complétion) le relâchement (*laxity* ou encore *slack time*) de chaque tâche est recalculé pour permettre d'exécuter en priorité la tâche la « moins relâchée » (c.-à-d. la plus contrainte). Le relâchement d'une tâche est calculé selon la formule $(R - t) - C'$ où R est l'échéance de la tâche, t est le temps réel écoulé dans cette itération et C' est le temps d'exécution restant pour cette itération. L'idée derrière cet algorithme est de réduire les changements de contexte (préemption) au minimum et en les retardant si nécessaire (ayant considéré les contraintes bien entendu).

- Q 3.3 Déterminez la séquence d'ordonnement du système avec l'algorithme LLF.
- Q 3.4 Comparez les temps de réponse des tâches et le temps de réponse moyen du système selon les algorithmes RM et LLF.

Exercice 4 *A new hope state*

Considérons un ordinateur embarqué dans lequel les processus partagent un espace mémoire comme seule ressource (autre que le processeur). Il s'agit d'une zone mémoire servant pour le diagnostic des pannes et contenant le journal des événements survenant sur le ordinateur. Chaque processus écrit des informations dans le journal.

Cette ressource n'est accessible qu'en accès exclusif et non requérable, c'est-à-dire qu'un accès en écriture lancé pour le compte d'un processus se termine normalement avant de pouvoir en lancer un autre. Un processus peut être en exécution, en attente du processeur (c.-à-d. prêt) ou bloqué à cause de l'écriture dans la mémoire.

L'état bloqué se divise en deux états : attente d'écriture (en cours d'écriture par un autre processus) et écriture en cours. Les demandes d'écriture sont gérées à l'ancienneté. L'écriture est réalisée par un périphérique, libérant ainsi le processeur pour un calcul (d'où libération de l'état d'exécution).

Dans ce système, on considère les quatre processus P1, P2, P3 et P4 pour lesquels on sait que :

- P1 et P2 sont des processus appartenant à la classe SCHED_FIFO ;
- P3 et P4 sont des processus appartenant à la classe SCHED_RR.

Les processus de la classe SCHED_FIFO sont toujours plus prioritaires que les processus de la classe SCHED_RR. Dans la classe SCHED_FIFO, le processeur est donné au processus de plus haute priorité. Ce dernier peut être préempté par un processus de la même classe ayant une priorité supérieure. Dans la classe SCHED_RR, le processeur est donné en premier au processus de plus haute priorité pour un quantum de temps égal à 10ms. La politique appliquée est ensuite celle du tourniquet.

Les 4 processus démarrent à $t=0s$ et ont le comportement suivant (la priorité la plus élevée est la plus importante) :

- P1 (priorité de 100) : calcul pendant 40ms, écriture pendant 50ms, calcul pendant 30ms, écriture pendant 40ms, calcul pendant 20ms ;
- P2 (priorité de 99) : calcul pendant 30ms, écriture pendant 80ms, calcul pendant 80ms, écriture pendant 20ms, calcul pendant 10ms ;
- P3 (priorité de 99) : calcul pendant 40ms, écriture pendant 40ms, calcul pendant 10ms ;
- P4 (priorité de 98) : calcul pendant 80ms.

Q 4.1 Déterminez le chronogramme des quatre processus.

Exercice 5 (optionnel)

En considérant un système exécutant les tâches τ_9 , τ_{10} , τ_{11} et τ_{12} définies dans le tableau ci-après, établissez le chronogramme d'exécution pour un algorithme d'ordonnancement à priorité fixe :

- Q 5.1 Dans le cas d'un SE non-préemptif ;
Q 5.2 Dans le cas d'un SE préemptif.

Tâche	Date d'activation (r)	Durée d'exécution (C)	Priorité
τ_9	0s	4s	3
τ_{10}	1,001s	3s	4
τ_{11}	2,001s	3s	6
τ_{12}	3,001s	5s	5

Note : un quantum de temps vaut 1ms, une valeur de priorité élevée indique une priorité importante.

q=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
τ_1															
τ_2															
τ_3															
τ_4															
τ_5															

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
τ_1															
τ_2															
τ_3															
τ_4															
τ_5															

q=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
τ_1															
τ_2															
τ_3															
τ_4															
τ_5															

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
τ_1															
τ_2															
τ_3															
τ_4															
τ_5															

q=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
τ_6																																		
τ_7																																		
τ_8																																		

q=0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
τ_6																																	
τ_7																																	
τ_8																																	

P1 0	100										200										300										400									
E en cours																																								
Att. d' E																																								
Ready																																								
Running																																								
P2																																								
E en cours																																								
Att. d' E																																								
Ready																																								
Running																																								
P3 0																																								
E en cours																																								
Att. d' E																																								
Ready																																								
Running																																								
P4 0																																								
E en cours																																								
Att. d' E																																								
Ready																																								
Running																																								