

Ingénieur Civil des Mines  
2ème année Conception Sûre des Systèmes Embarqués et Ambiants – Module SI342  
2005-2006

22 mars 2006

Ordonnancement sous contraintes temps réel

---

---

Pour vérifier la solution proposée, lorsque cela est possible, on pourra utiliser le logiciel écrit par *Mathieu Grenier* (projet TRIO – LORIA) et disponible à l'adresse :

<http://webloria.loria.fr/equipes/TRIO/simulateur/SimApplet.html>

Ce logiciel calcule si un ensemble de tâches périodiques, sous contraintes d'échéance, est ordonnançable. La stratégie d'ordonnancement et repose sur :

- un partitionnement de l'ensemble des tâches en niveaux (*layers*) numérotés 1, 2, ... (la priorité des tâches au niveau  $i$  est supérieure à la priorité des tâches de niveau  $j$  si  $i < j$ )
- sur chaque niveau, une *politique donnée d'ordonnancement* (affectation de priorités dynamique ou statique ; politique préemptive ou non préemptive) non oisive (c'est-à-dire, s'il reste du travail en attente, le processeur est occupé)

Le modèle d'une tâche,  $t_k$ , est défini par :

- son temps d'exécution  $C_k$ ,
  - sa période  $T_k$ ,
  - son échéance relative  $D_k$ ,
  - son "Offset", c'est-à-dire la date d'activation de la première instance de cette tâche  $A_k$ .
- 
- 

### Exercice 1

Soient 2 tâches  $T_1$  et  $T_2$ , périodiques, indépendantes, à départ simultané :

Tâches	Période	Temps d'exécution	Echéance
T1	2	1	2
T2	3	1	1

On considère une politique préemptive.

Cette configuration est-elle ordonnançable selon l'algorithme « Rate Monotonic » ?

Cette configuration est-elle ordonnançable selon l'algorithme « Deadline Monotonic » ?

## Exercice 2

Soient 4 tâches  $T_1$ ,  $T_2$ ,  $T_3$  et  $T_4$ , périodiques, indépendantes, à départ simultané et à échéance sur requête :

Tâches	Période	Temps d'exécution
T1	7	2
T2	5	1
T3	7	1
T4	10	1

On considère une politique préemptive.

Cette configuration est-elle ordonnançable ?

Donner une séquence produite FPP pour une affectation statique de priorité « Rate Monotonic » ?

## Exercice 3

La configuration suivante est ordonnançable selon l'algorithme d'affectation de priorité « Earliest Deadline First » ; pourquoi ? Donner la séquence produite par cet algorithme.

Est-elle ordonnançable selon la politique « Rate Monotonic » ? Si oui, donner la séquence construite et identifier les différences avec la politique précédente.

Les tâches sont à départ simultané.

Dans les deux cas, on considère une politique préemptive.

Tâches	Période	Temps d'exécution	Echéance
T1	3	2	3
T2	7	2	6

## Exercice 4

La configuration suivante est-elle ordonnançable selon l'algorithme « Earliest Deadline First » . Si oui, donner la séquence produite par cet algorithme.

Les tâches sont à départ simultané et la politique est préemptive.

Tâches	Période	Temps d'exécution	Echéance
T1	2	1	2
T2	3	1	3
T3	5	1	5

## Exercice 5

La configuration suivante est ordonnable selon l'algorithme « Earliest Deadline First » (voir ci-dessus) . Si oui, donner la séquence produite par cet algorithme.

Tâches	Période	Temps d'exécution	Echéance	Offset
T1	3	1	3	4
T2	5	3	5	0

## Exercice 6

La configuration suivante est ordonnable selon l'algorithme « Earliest Deadline First ». Donner son ordonnancement selon la politique « Least Laxity First » (attention, LLF n'est pas géré par le logiciel fourni).

Comparer le nombre de commutations de contexte induites par les algorithmes « Earliest Deadline First » et « Least Laxity First ».

Tâches	Période	Temps d'exécution	Echéance
T1	3	2	3
T2	7	2	6

Les tâches sont à départ simultané et la politique est préemptive

## Exercice 7

Soient 3 tâches  $T_1$ ,  $T_2$  et  $T_3$ , périodiques, indépendantes, à échéance sur requête et à départ simultané. La politique est préemptive :

Tâches	Période	Temps d'exécution
T1	5	1
T2	6	2
T3	15	4

- On peut affirmer que cette configuration est ordonnable suivant la stratégie « Earliest Deadline First » (EDF). Pourquoi ?
- Donnez le diagramme modélisant l'occupation du CPU sur une durée suffisante. Faites apparaître sur ce diagramme les instants et les raisons de l'appel de l'algorithme de choix dans le cas d'un ordonnancement de ces tâches suivant EDF.
- Peut-on dire si cette configuration est ou non ordonnable suivant la stratégie « Rate Monotonic » ? Quels moyens devrait-on mettre en œuvre pour obtenir l'un ou l'autre verdict ?
- On suppose qu'une tâche apériodique, AP, co-existe avec cette configuration. Son utilisation du processeur est au mieux de 4 unités de temps et au plus de 12 unités de temps. Cette tâche apériodique est activée par un événement externe E. L'exécution de chaque instance de AP doit être terminée avant la création de l'instance suivante. Donnez une condition suffisante sur les occurrences de E pour que la propriété d'échéance de AP soit toujours vérifiée. Proposez des solutions pour l'ordonnancement des tâches apériodiques contraintes par une échéance et donner la séquence obtenue.

## Exercice 8

Soient 3 tâches  $T_1$ ,  $T_2$  et  $T_3$ , périodiques, indépendantes, à départ simultané. La politique est préemptive

Tâches	Période	Temps d'exécution	Echéance
T1	20	3	7
T2	5	2	4
T3	10	1	8

Proposer des solutions pour l'ordonnancement de tâches aperiodiques et donner les séquences produites par chaque solution pour l'arrivée de tâches aperiodiques suivante

Tâches	Date d'arrivée	Temps d'exécution	Echéance
Tap1	24	2	6
Tap2	10	1	6
Tap3	11	2	7

## Exercice 9

Soient 4 tâches  $T_1$ ,  $T_2$ ,  $T_3$  et  $T_4$ , périodiques, indépendantes ordonnancées selon deux niveaux d'ordonnancement  $L_1$ , regroupant  $T_1$  et  $T_2$ , et  $L_2$ , regroupant  $T_3$  et  $T_4$ . On considère que, sur  $L_1$ , la politique d'affectation des priorités est RM

Les politiques sur chaque niveau sont préemptives

Tâches	Période	Temps d'exécution	Echéance
T1	10	2	10
T2	8	1	8

On considère que, sur  $L_1$ , la politique d'affectation des priorités est EDF

Tâches	Période	Temps d'exécution	Echéance
T1	20	2	20
T2	5	1	5

Simuler cette configuration et commenter les chronogrammes obtenus.

## Exercice 10

On considère un algorithme non-préemptif, non oisif, à affectation de priorité dynamique, par exemple EDF.

Tâches	Période	Temps d'exécution	Echéance relative	Offset
T1	12	3	5	2
T2	12	4	10	1
T3	12	3	12	0

Donner le chronogramme de cette configuration. Les temps d'exécution sont considérés comme des "pires cas". Il est donc possible qu'une instance d'une tâche  $T_i$  se termine avant  $C_i$ . Donner le chronogramme de la configuration si on considère que la première instance de la tâche  $T_3$  se termine au bout d'une unité. Conclure sur cet exemple.