



UNIVERSITE L'ARBI BEN M'HIDI OUM EI BOUAGHI  
INSTITUT DES SCIENCES ET DES TECHNIQUES APPLIQUÉES  
DÉPARTEMENT réseaux et télécommunications

1<sup>ère</sup> Année Master SRT

Année Universitaire 2024/2025

Systèmes Embarqués Temps Réel (SETR)

Examen (1h30min)

Questions de cours (06 pts)

**Question 1** Expliquez les types des systèmes en temps réel ? donner des exemples pour chaque type ?

**Question 2.** Donner une définition convenable pour tous ceux qui : tâche, monotâche, multitâche, processus, ordonnancement, RM, DM, LLF et EDF. ?

**Question 3** Donner une définition de systèmes embarqués Temps réel ? Expliquez avec des exemples ?

Exercice 1 (06)

Un système en temps réel comprend trois (3) tâches dont les caractéristiques sont :

Tache	Date de réveil $r_i$	Capacité $C_i$	Echéance $D_i$	Période $P_i$
T1	0	3	15	20
T2	0	40	60	80
T3	0	10	35	40

a) En utilisant les algorithmes d'ordonnancement « RM et DM », déterminer la priorité de chaque tâche, Calculer la période étude, et Temps libre processeur ?

b) Que peut-on apprendre du test d'ordonnancement basé sur le facteur d'utilisation ?

c) Tracer un diagramme de GANTT et vérifier que les échéances sont respectées ?

Exercice 2 (08)

Etant donné un entier positif  $C_2$ , nous considérons le système de tâches  $\{T1, T2, T3\}$ , avec  $T1 = (C1=2, 9)$ ,  $T2 = (C2, 6)$  et  $T3 = (C3=1, 9)$ . Sachant que  $P_i=D_i$ ,  $r_i=0$ .

1. Déterminez le temps d'exécution  $C_2$  de tâche  $T2$  pour que le système est ordonnançable par l'algorithme RM, puis celui de EDF.

2. Tracer le digramme de GANTT par l'algorithme EDF ?

3. Nous souhaitons à présent introduire plusieurs copies de la tâche  $T3$ ,  $C3$ .

a) Déterminez la condition obtenue sur  $C_2$ , en fonction de  $C_3$ , pour que ce nouveau système ordonnançable par LLF.

b) Calculez les valeurs numériques obtenues pour  $C_3$  variant de 1 à 5.

c) Tracer le digramme de GANTT par l'algorithme LLF pour  $C_3=4$  ?

**Req** : pour tracer le digramme de GANTT par EDF et LLF, calculer la période d'étude, Temps libre processeur

## Réponse

### Questions de cours (07 pts)

**Question 1** Expliquez les types des systèmes en temps réel ? donner des exemples pour chaque type ?

#### Réponse 1

### **1. Temps réel dur (hard real-time)**

Le non-respect d'une échéance d'une fonction ayant une contrainte temps réel dure est considéré comme une défaillance pouvant entraîner des conséquences catastrophiques.

#### **1.1 Exemples**

- Système ABS d'un véhicule,
- Déclencheur des Airbags,
- Contrôle du trafic aérien,
- Système de guidage d'un missile,
- Système de surveillance de centrale nucléaire.

### **2. Temps réel strict (firm real-time)**

Certaines fonctions ayant des contraintes temps réel souples doivent respecter un temps de réponse moyen tout en restant à l'intérieur d'une fenêtre temporelle stricte.

#### **2.1 Exemple**

- Fonction d'échantillonnage périodique

### **3. Temps réel lâche ou mou (soft real-time)**

Le non-respect d'une échéance d'une fonction temps réel à contrainte souple n'est pas catastrophique et la qualité de la réponse, bien que dégradée, reste acceptable. Un système temps réel souple ne comporte que des fonctions ayant des contraintes temps réel souples.

#### **3.1 Exemple**

- C'est le cas par exemple de fonctions de communication ou de traitement multimédia pour lesquels la qualité générale des documents n'est pas affectée de manière significative par le retard ou le manque d'informations d'une image ou d'un son (décalage entre le son et l'image...).

**Question 2.** Donner une définition convenable pour tous ceux qui : tâche, monotâche, multitâche, processus, ordonnancement, RM, DM, LLF et EDF. ?

#### Réponse 2

**Tâche.** Une tâche est une unité active de l'application, il s'agit d'un ensemble d'instructions séquentielles, ou bien un programme ou une partie de programme en exécution, il existe deux types de tâche

**Tâche immédiate (TI)** (hard, routine d'interruption) s'exécute lors de l'arrivée d'une interruption sur le processeur, sont les tâches les plus prioritaires d'une application, et sont en général liées aux entrées sorties

et doivent être les plus rapides possible. Ce sont elles qui s'exécutent à l'arrivée d'un événement dans le système. Elles ont pour rôle de prendre en charge le traitement de cet événement.

**Tâches différées (soft) :** Les tâches dites différées (TD), ou encore « logicielles », sont les tâches les moins prioritaires d'une application. Leur rôle est d'effectuer des opérations cycliques et générales.

**Monotâche,** A tout instant, un seul programme est exécuté ; un autre programme ne démarrera, sauf conditions exceptionnelles, que lorsque le premier sera terminé

**Multitâche,** : plusieurs processus (i.e. un « programme » en cours d'exécution) peuvent s'exécuter simultanément (systèmes multiprocesseurs) ou en quasi- parallélisme (systèmes à temps partagé)

**Processus,** C'est une entité dynamique qui naît, qui vit et qui meurt, par opposition à la notion de programme qui est une entité statique qui occupe un espace en mémoire ou sur disque et qui, tant qu'elle n'est pas exécutée, ne produit aucune action. D'autre part, un programme peut être vu comme un ensemble de modules (sous-programmes) et l'exécution d'un module donne naissance à un processus.

**Ordonnancement,** la gestion des tâches

**L'algorithme RM,** Est un algorithme à priorité fixe. Cet algorithme affecte des priorités aux tâches en fonction de leur période : plus leur période est petite, plus la tâche est prioritaire.

**L'algorithme DM,** est basé les échéances relatives (Délai critique) des tâches. Les priorités des tâches sont inversement proportionnelles aux délais critiques des tâches, La tâche la plus prioritaire est celle ayant la plus petite Délai critique.

**L'algorithme LLF,** Choisit d'exécuter en priorité la tâche possédant la laxité dynamique la plus faible, ce qui se traduit de la manière suivante : à chaque instant  $t$ , la tâche la plus prioritaire est celle dont la laxité du processeur à  $t$ , définie par  $L(t) = r_i + D_i - (t + C_i(t))$  ou  $(L = D - R)$ , est la plus petite.

**L'algorithme EDF.** Le principe d'allocation de priorités dans cet algorithme est que la tâche dont l'échéance absolue arrive le plus tôt aura la priorité la plus élevée. Les priorités sont réévaluées à chaque fois que l'ordonnanceur doit déterminer la prochaine tâche à exécuter.

$$\text{échéance absolue } d_i = D_i + r_i$$

**Question 3** Donner une définition de systèmes embarqués Temps réel ? Expliquez avec des exemples ?

Le système embarqué c'est l'échange entre la partie hardware et la partie software

**Défi. 1.** Le temps réel est le pilotage à tout instant d'un système évolutif dans le temps

**Défi.2.** Un système temps réel est un système pour lequel le temps d'arrivée du résultat est aussi important que le résultat lui-même.

Le temps réel est lié à la notion de pilotage d'un système soumis à des contraintes de temps. On ne peut pas désigner de système temps réel, un calculateur utilisé pour des applications de gestion ou des applications scientifiques ; les contraintes de temps au quels sont soumis ces systèmes sont généralement liées à la

productivité souhaitée. Un système embarqué à bord d'un avion a pour but de fournir la position instantanée de l'avion par rapport à un point de référence terrestre. Le temps de traitement global correspondant au travail à réaliser est finalement le temps mis pour :

- Acquérir toutes les données d'entrée en provenance des capteurs, Traiter effectivement ces données,
- Fournir le résultat correspondant.

Notons  $T_1$  le temps d'acquisition,  $T_2$  le temps de traitement des données et  $T_3$  le temps nécessaire à la présentation des résultats. Il est impératif que l'inéquation suivante soit respectée pour garantir un fonctionnement correct de l'application temps réel :

$$Tl = T_1 + T_2 + T_3 \quad (Tl : \text{temps limite de production du résultat})$$

Il ne reste plus qu'à connaître avec précision la valeur de  $Tl$  au-delà de laquelle le résultat fourni, bien que juste, ne présente plus d'intérêt. En effet, au-delà de ce temps limite, le résultat obtenu n'est plus en rapport avec les conditions présentes dans lesquelles évolue Le système.

Par exemple, que dirait-on si le système de calcul de la position d'un avion fournissait un résultat, au demeurant correct par rapport aux données d'entrées, au bout de plusieurs dizaines de minutes ? Dans notre exemple, il est raisonnable de penser que la valeur de  $Tl$  n'est que de l'ordre de quelques millisecondes. Il peut d'ailleurs être conditionné par d'autres paramètres s'appliquant à l'avion comme sa vitesse instantanée.

### Rép.Exercice 1 (06)

1. **RM** Plus leur période est petite, plus la tâche est prioritaire

**DM** La tâche la plus prioritaire est celle ayant la plus petite Délai critique.

#### Période d'étude

20 : 20, 40, 60, 80

40 : 40, 80

80 : 80, 160

PPCM( $T_i$ )=80

#### Temps libre

$$T_{\text{libre}} = (1 - U) * H = \left(1 - \sum_{i=1}^n \frac{C_i}{P_i}\right) * PPCM\{T_i\} \quad i \in [1, n] \quad \text{Pour DM et RM}$$

Facture de charge pour RM et DM  $P_i \neq D_i$

Tache	Date de réveil $r_i$	Capacité $C_i$	Echéance $D_i$	Période $P_i$
T1	0	3	15	20
T2	0	40	60	80
T3	0	10	35	40

#### Pour RM

$$\left(\sum_{i=1}^n \frac{C_i}{P_i}\right) = 3/20 + 40/80 + 10/40 = (12 + 40 + 20) / 80 = 72/80 = 0.9 > 0.77$$

$$T_{libre\ RM} = (1 - 0.9) * 80 = 8$$

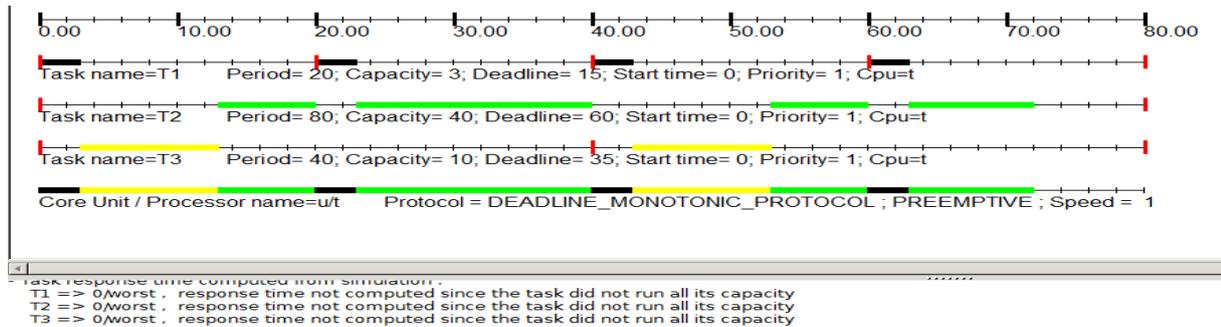
### Pour DM

$$\left(\sum_{i=1}^n \frac{C_i}{D_i}\right) = (3/15 + 40/60 + 10/35) = 0.2 + 0.66 + 0.28 = 1.14 > 0.77$$

2. Pour les deux algorithmes le système n'est pas ordonnançable

### 3. Diagramme de GANTT

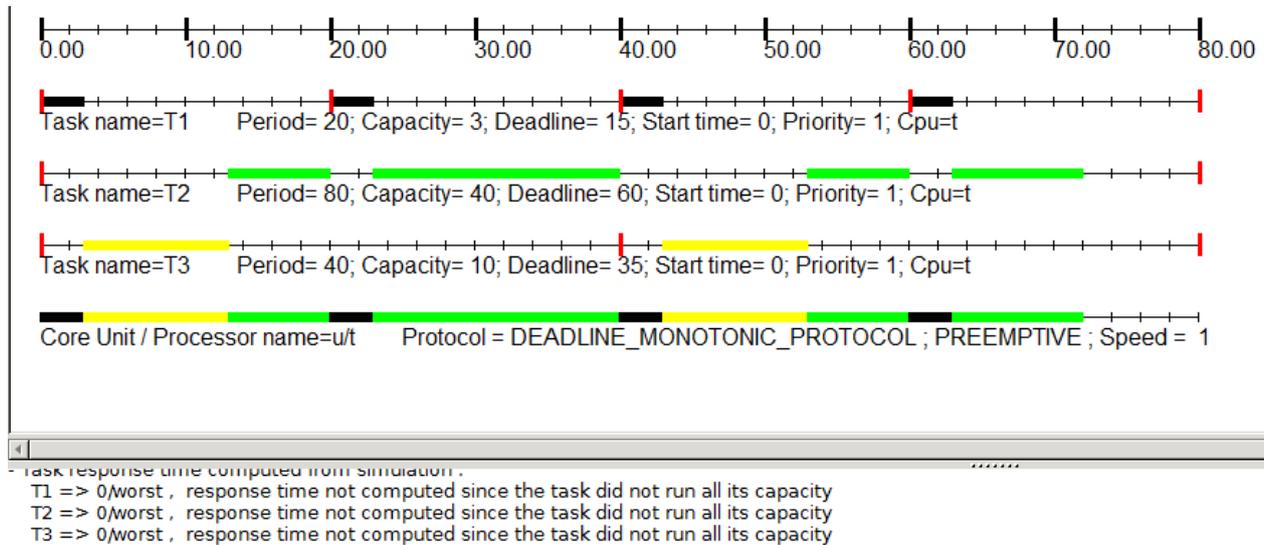
#### 3.1 Pour RM



#### Scheduling simulation, Processor t :

- Number of context switches : 9
- Number of preemptions : 3
- Task response time computed from simulation :
  - T1 => 3/worst
  - T2 => 72/worst, missed its deadline (absolute deadline = 60 ; completion time = 72)
  - T3 => 13/worst
- Some task deadlines will be missed : the task set is not schedulable.

#### 3.2 Pour DM



#### Scheduling simulation, Processor t :

- Number of context switches : 9
- Number of preemptions : 3
- Task response time computed from simulation :
  - T1 => 3/worst
  - T2 => 72/worst, missed its deadline (absolute deadline = 60 ; completion time = 72)
  - T3 => 13/worst
- Some task deadlines will be missed : the task set is not schedulable.

### Rép.Exercice 2 (08)

1.

**Ordonnabilité RM**

$2/9 + c2/6 + 1/9 \leq 0,78$  donc  $c2 \leq 2,68$

**Ordonnable EDF**

$2/9 + c2/6 + 1/9 \leq 1$  donc  $c2 \leq 4$

2. Diagramme de GANTT Par EDF.

PPCM=18

9 9 18

6 6 12 18

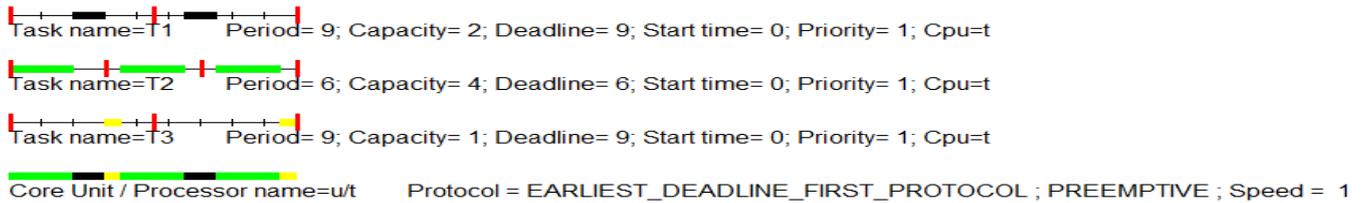
9 9 18

$$Tlibre = (1 - U) * H = \left(1 - \sum_{i=1}^n \frac{C_i}{P_i}\right) * PPCM\{T_i\} \quad i \in [1, n]$$

$U = 2/9 + 4/6 + 1/9 = 0.22 + 0.66 + 0.11 = 0.99 \leq 1$

**Tlibre** =  $(1 - 0.99) * 18 = 0.18$  alors le temps libre processeur n'existe plus sur 18 unités

**Vérification par cheddar**



```

t1 => 3/worst
t2 => 72/worst . missed its deadline (absolute deadline = 60 ; completion time = 72)
t3 => 13/worst
- Some task deadlines will be missed : the task set is not schedulable.
    
```

**Scheduling simulation, Processor t :**  
 - Number of context switches : 6  
 - Number of preemptions : 0  
 - Task response time computed from simulation :  
 T1 => 6/worst  
 T2 => 5/worst  
 T3 => 9/worst  
 - No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

3. a pour C3 tache T3 LLF

$$U = \sum_{i=1}^n \frac{C_i}{P_i} \leq 1$$

$2/9 + C2/6 + C3/9 \leq 1$

$$C_2 \leq \frac{14 - 2C3}{3}$$

b.

Pour C3=1 donc C2≤4

Pour C3=2 donc C2≤3.33

Pour C3=3 donc C2≤2.66

Pour C3=4 donc C2≤2

Pour C3=5 donc C2≤1.33

C. le diagramme de GANTT par l'algorithmme LLF pour C3=4

Pour C3=4 donc C2=2

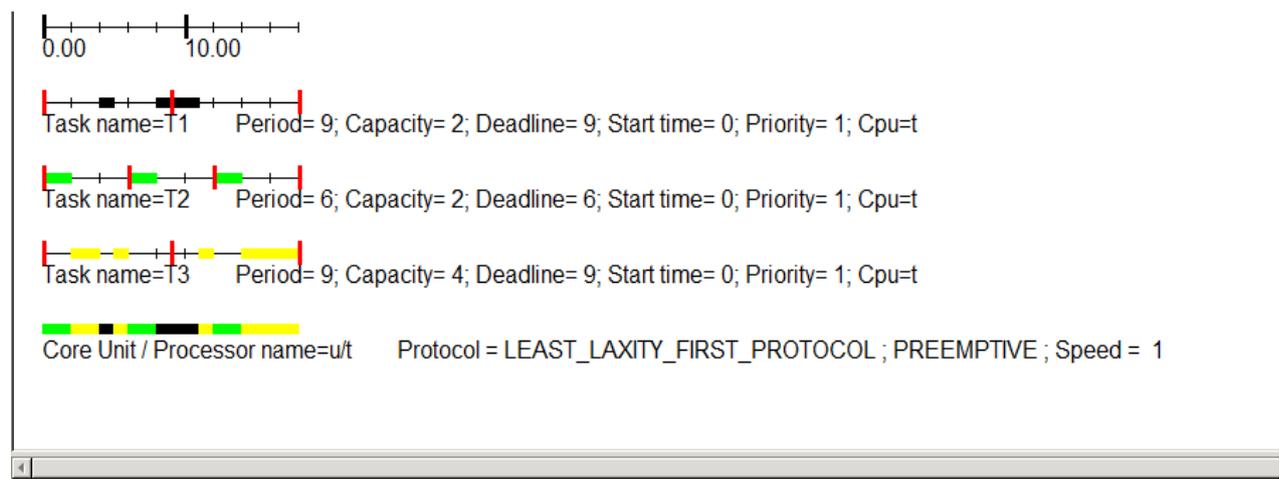
T1(0,2,9,9), T2(0,2,6,6), T3(0,4,9,9)

$$Tlibre = (1 - U) * H = \left(1 - \sum_{i=1}^n \frac{C_i}{P_i}\right) * PPCM\{T_i\} \quad i \in [1, n]$$

$$0.11 + 0.33 + 0.44 = 0.88$$

$$Tlibre_{LLF} = (1 - 0.71) * 18 = 2$$

### Vérification Par LLF



T1 => 0/worst  
T2 => 1/worst  
T3 => 8/worst  
- No deadline missed in the computed scheduling : the task set is schedulable if you computed the scheduling on the feasibility interval.

#### Scheduling simulation, Processor t :

- Number of context switches : 8  
- Number of preemptions : 3

- Task response time computed from simulation :

T1 => 9/worst

T2 => 2/worst

T3 => 12/worst , missed its deadline (absolute deadline = 9 ; completion time = 12)

- Some task deadlines will be missed : the task set is not schedulable.