

Les Systèmes Temps Réel

Chapitre 2: Ordonnancement des tâches périodiques

Cours Master 1 GLSD

2019/2020

Pr. Salim BITAM

Département d'Informatique, Université de Biskra

07000 Biskra, Algérie

Les paramètres de temps (1)

- r: la date de réveil
- C: la durée d'exécution maximale
- D: le délai critique maximum acceptable de pour exécuter une tâche
- P: la période d'activation d'une requête (pour les tâches périodique)
- Il est à noter que : $C \leq D \leq P$

Les paramètres de temps (2)

- d : l'échéance ($= r+D$)
- U : le facteur d'utilisation du processeur
- $U = C/P$
- Ch : le facteur de charge du processeur
- $Ch = C/D$
- s : la date de début d'exécution de la tâche
- e : la date de fin d'exécution de la tâche

Les paramètres de temps (3)

- $D(t) = d - t$: le délai critique résiduel à la date t ($0 \leq D(t) \leq D$)
- $C(t)$: la durée d'exécution résiduelle à la date t ($0 \leq C(t) \leq C$)
- $L = D - C$: la laxité nominale de la tâche ; c'est le retard maximum pour son début d'exécution s (si elle est seule)
- $L(t) = D(t) - C(t)$: la laxité nominale résiduelle
(retard maximum pour reprendre l'exécution)
- $TR = e - r$: le temps de réponse de la tâche
- $CH(t) = C(t)/D(t)$: la charge résiduelle ($0 \leq CH(t) \leq C/P$)

Remarques

- Pour une tâche périodique : $r_k = r_0 + k * P$
- si $D = P$, la tâche est définie à échéance sur requête

La configuration d'ordonnancement (1)

- Une configuration : ensemble de n tâches à ordonnancer
- Les départs échelonnés ou simultanés

- Le facteur d'utilisation du processeur $U = \sum_{i=1}^n \frac{C_i}{P_i}$

- Le facteur de charge $CH = \sum_{i=1}^n \frac{C_i}{D_i}$

La configuration d'ordonnancement (2)

- L' intervalle d'étude : intervalle de temps minimum pour assurer l'ordonnançabilité d'une configuration
- L' intervalle d'étude = le PPCM des périodes dans le cas d'une configuration de tâches périodiques

Ordonnancement des tâches périodiques indépendantes : Introduction

- Tâches indépendantes:
 - pas de partage de ressources
 - pas de contraintes de précédence
- cas des algorithmes en ligne dynamique
 - sur la base d'une priorité définie sur la base
 - d'un paramètre temporel de la tâche
- priorité constante ou variable
- Parfois l'ordonnancement est à un test d'acceptabilité

Algorithmes d'ordonnancement en ligne à priorité constante

- **Rate Monotonic Analysis (RMA):**
- **Principe:** la tâche de plus petite période est la plus prioritaire
- **Un test d'acceptabilité:** sur le facteur d'utilisation :
(condition suffisante)

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

- **Important:** RMA est applicable seulement sur les tâches à échéance sur requête

Exemple:

- Soit la configuration suivante:
 - T1 ($r_0 = 0$, $C=3$, $P=20$),
 - T2 ($r_0 = 0$, $C=2$, $P=5$),
 - T3 ($r_0 = 0$, $C=2$, $P=10$)
-
- Correction: Voir le chronogramme donné dans les première séance

Algorithmes d'ordonnancement en ligne à priorité constante

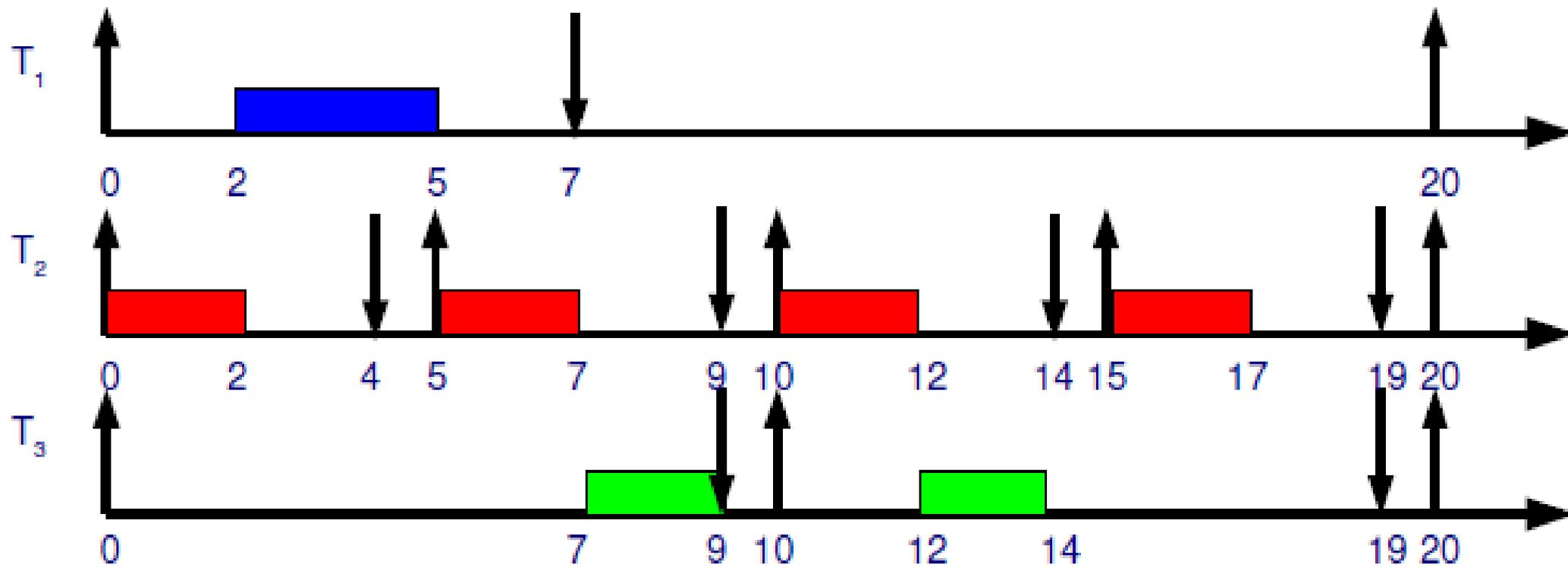
- **Deadline Monotonic Analysis (DMA):**
- **Principe:** la tâche de plus petit délai critique est la plus prioritaire
- **Un test d'acceptabilité:** sur le facteur de charge :
(condition suffisante)

$$CH = \sum_{i=1}^n \frac{C_i}{D_i} \leq n(2^{1/n} - 1)$$

- **Important:** DMA est applicable pour toutes les configurations

Exemple:

- Soit la configuration suivante:
- T1 ($r_0 = 0$, $C=3$, $D=7$, $P=20$),
- T2 ($r_0 = 0$, $C=2$, $D=4$, $P=5$),
- T3 ($r_0 = 0$, $C=2$, $D=9$, $P=10$)



Algorithmes d'ordonnancement en ligne à priorité variable

- **Earliest Deadline First (EDF):**
- **Principe:** la tâche de plus proche échéance est la plus prioritaire
- **Un test d'acceptabilité:**
 1. Pour les configurations dont toutes les tâches sont définies échéance sur requête, le test est sur le facteur d'utilisation :
(condition nécessaire et suffisante)

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

Algorithmes d'ordonnancement en ligne à priorité variable : EDF(2)

2. Pour les configurations dont la présence d'une tâche qui n'est pas définie échéance sur requête, le test est :

- sur le facteur d'utilisation :

(condition nécessaire)

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

- sur le facteur de charge

(condition suffisante)

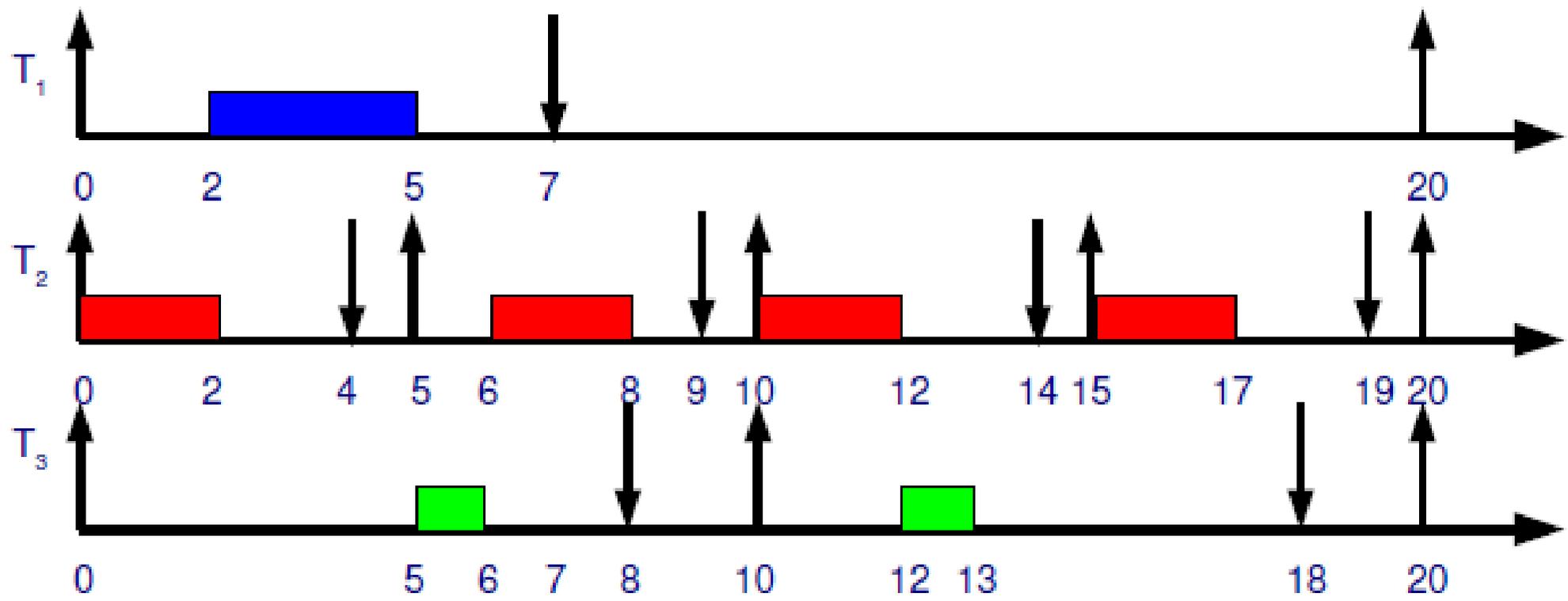
$$CH = \sum_{i=1}^n \frac{C_i}{D_i} \leq 1$$

EDF: Exemple

- Soit la configuration suivante:
- T1 ($r_0 = 0, C=3, D=7, P=20$),
- T2 ($r_0 = 0, C=2, D=4, P=5$),
- T3 ($r_0 = 0, C=1, D=8, P=10$)

$$U = \sum_{i=1}^n \frac{C_i}{P_i} = 0,65 \text{ (condition nécessaire vérifiée)}$$

$$CH = \sum_{i=1}^n \frac{C_i}{D_i} = 1,05 \text{ (condition suffisante non vérifiée)}$$



Algorithmes d'ordonnancement en ligne à priorité variable

- **Least Laxity First (LLF):**
- **Principe:** la tâche qui possède la plus petite laxité dynamique est la plus prioritaire
- **Un test d'acceptabilité:**
 1. Pour les configurations dont toutes les tâches sont définies échéance sur requête, le test est sur le facteur d'utilisation :
(condition nécessaire et suffisante)

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

Algorithmes d'ordonnancement en ligne à priorité variable : LLF(2)

2. Pour les configurations dont la présence d'une tâche qui n'est pas définie échéance sur requête, le test est :

- sur le facteur d'utilisation :

(condition nécessaire)

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

- sur le facteur de charge

(condition suffisante)

$$CH = \sum_{i=1}^n \frac{C_i}{D_i} \leq 1$$

LLF: Exemple

- Soit la configuration suivante:
- T1 ($r_0 = 0$, $C=3$, $D=7$, $P=20$),
- T2 ($r_0 = 0$, $C=2$, $D=4$, $P=5$),
- T3 ($r_0 = 0$, $C=1$, $D=8$, $P=10$)

$$U = \sum_{i=1}^n \frac{C_i}{P_i} = 0,65 \text{ (condition nécessaire vérifiée)}$$

$$CH = \sum_{i=1}^n \frac{C_i}{D_i} = 1,05 \text{ (condition suffisante non vérifiée)}$$

- Calcul de la laxité dynamique: exemple

✓ $t=0$

$$x \ L_1 = 7-3 = 4$$

$$x \ L_2 = 4-2 = 2$$

$$x \ L_1 = 8-1 = 7$$

