Programmation Temps Réel

Tâches Temps Réel

Yann Thoma

Reconfigurable and Embedded Digital Systems Institute
Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud

Septembre 2017

Introduction

- Une tâche vise un traitement particulier
  - Gestion d'un capteur (ou plusieurs)
  - Gestion d'un actuateur (ou plusieurs)
  - Calcul particulier
  - Logging
  - ...
- La décomposition en tâches est naturel
- Chaque tâche peut disposer de contraintes particulières
  - Période, échéance, criticité
- Les définitions données seront indépendantes
  - De la présence d'un noyau multi-tâche
  - Du type d'ordonnancement

Types de tâches

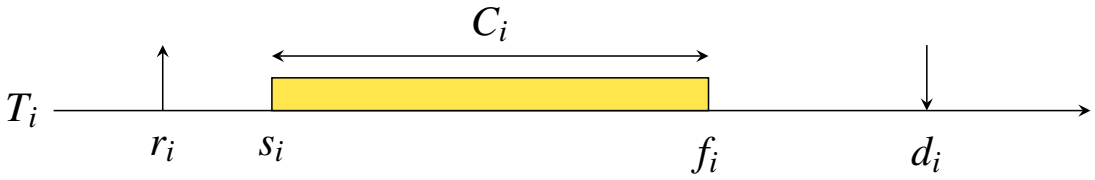
- Tâches périodiques
  - La tâche se répète régulièrement
  - Elle possède une période
  - Exemple:
    - Monitoring
    - Lecture de capteurs
- Tâches apériodiques
  - La tâche peut s'exécuter n'importe quand
  - Exemple:
    - Alarmes
    - Interaction avec un utilisateur
- Tâches sporadiques
  - Correspond à une tâche apériodique dont l'intervalle de temps minimal séparant deux occurences est connu

Dépendances

- Tâches indépendantes
  - Les tâches peuvent être exécutées dans un ordre quelconque
- Tâches dépendantes
  - Les tâches sont reliées entre elles par des contraintes de dépendances
  - Par exemple, une tâche T_1 doit attendre que T_0 se termine pour s'exécuter
- L'ordonnancement des tâches doit prendre en compte ce critère

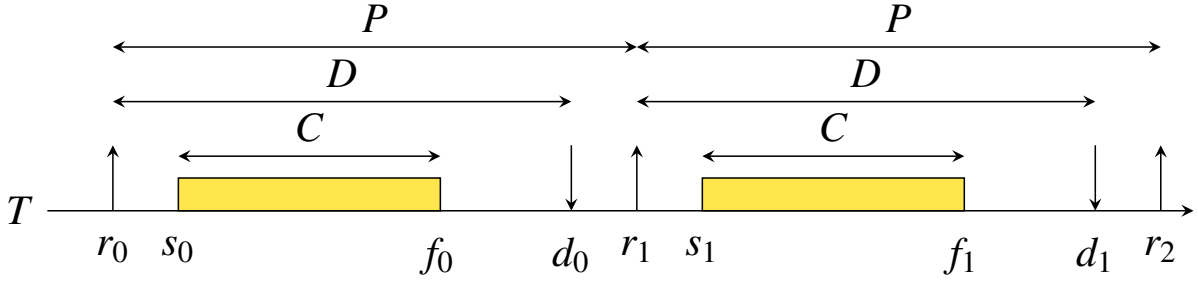
Paramètres d'une tâche

- r: la date de réveil de la tâche (ou date de demande d'activation)
- C: sa durée d'exécution, calculée en temps processeur. Il s'agit du pire temps d'exécution.
- d: son échéance, au-delà de laquelle le résultat est jugé comme étant non pertinent
- s: date de début d'exécution de la tâche
- f: date de fin d'exécution de la tâche



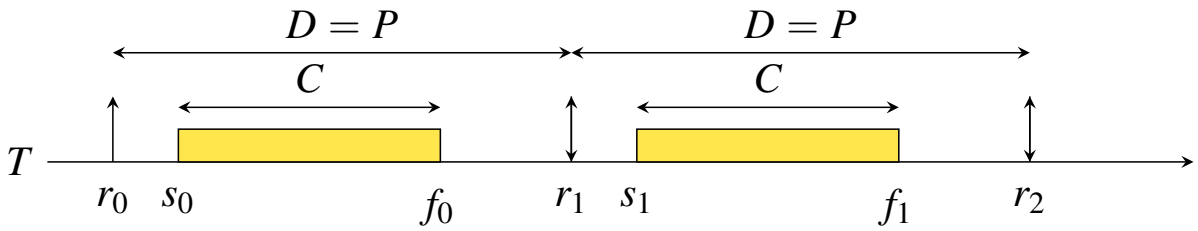
Paramètres d'une tâche périodique

- D: son délai critique, au-delà duquel le résultat est jugé comme étant non pertinent
- P: sa période, pour le cas d'une tâche périodique
- d: pour une tâche à contraintes strictes, son échéance, calculée comme étant $d = r + D$



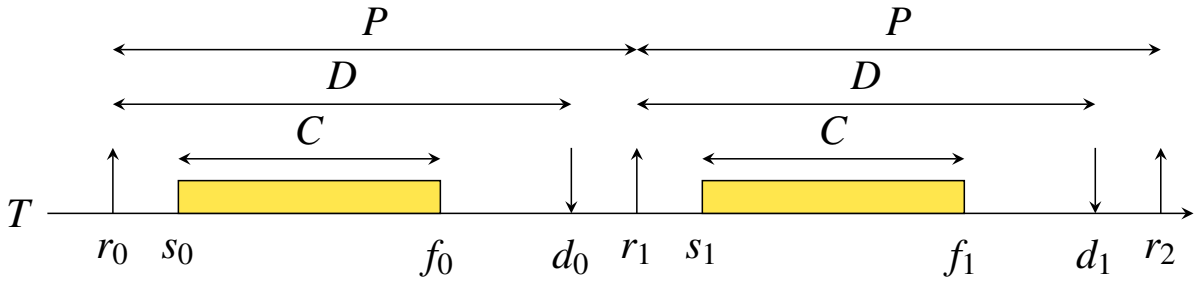
Paramètres d'une tâche périodique à échéance sur requête

- $D = P$: son délai critique, au-delà duquel le résultat est jugé comme étant non pertinent



Paramètres : Charge et utilisation

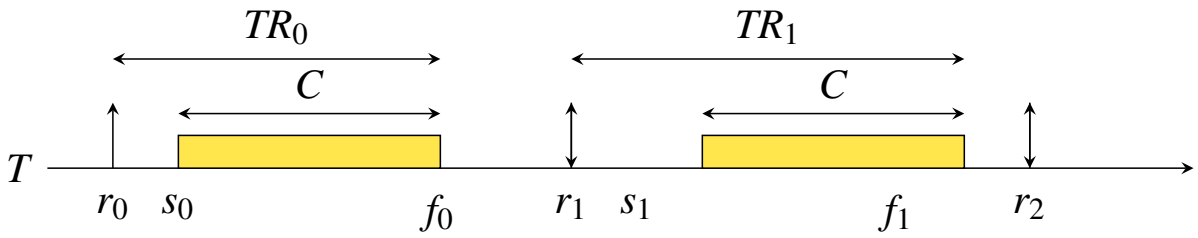
- $u = \frac{C}{P}$: son facteur d'utilisation du processeur
- $ch = \frac{C}{D}$: son facteur de charge du processeur



- $C = 4, D = 7, P = 8$
- $u = \frac{4}{8} = 0.5$
- $ch = \frac{4}{7} \cong 0.57$

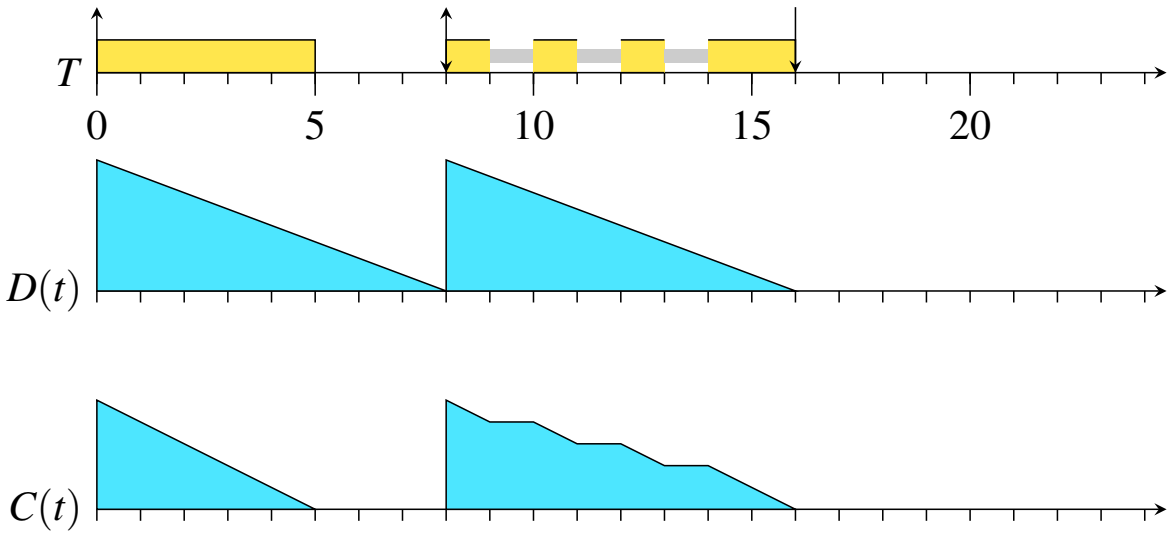
Paramètres: Temps de réponse

- $TR_i = f_i - r_i$: son temps de réponse. L'échéance est respectée si $TR_i \leq D$



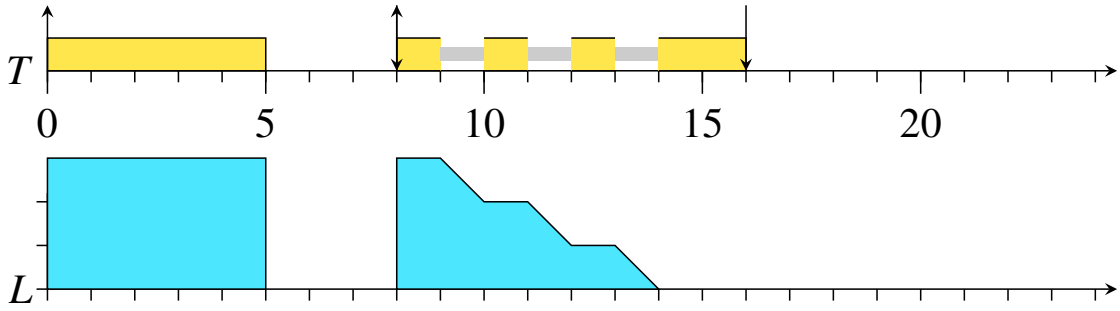
Paramètres: Délai critique

- $D(t) = d - t$: son délai critique résiduel au temps t
- $C(t)$: sa durée d'exécution résiduelle au temps t
- Exemple: Une tâche d'échéance 8 et coût 5.

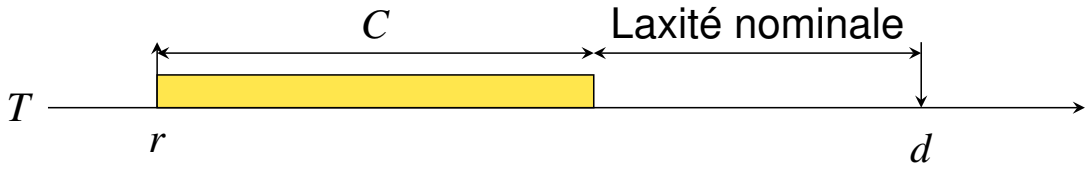


Paramètres: Laxité

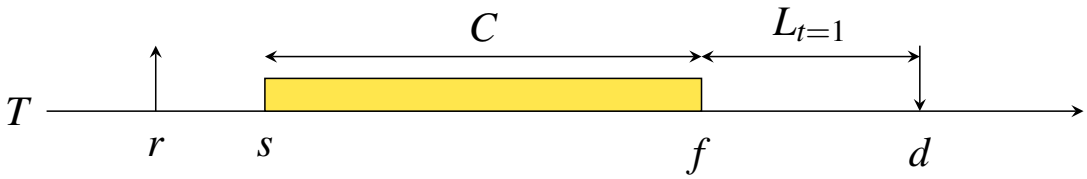
- $L = D - C$: sa laxité nominal. Indique le retard maximum que peut prendre la tâche sans dépasser son échéance
- $L(t) = D(t) - C(t)$: sa laxité résiduelle au temps t
- Exemple: Une tâche d'échéance 8 et coût 5. La laxité $L = 8 - 3 = 5$
- La laxité résiduelle est, sur cet exemple d'exécution:



Laxité



- Scénario 1:



- Scénario 2:

Propriétés des tâches

Nous pouvons noter les propriétés suivantes:

- $u \leq 1$
- $ch \leq 1$
- $0 \leq D(t) \leq D$
- $0 \leq C(t) \leq C$
- $L(t) = D(t) - C(t) = D + r - t - C(t)$

Gigue

- La guigue (*jitter* en anglais) correspond à une variation de latence sur un signal. Typiquement dans le cadre du temps réel il s'agit de la variation de temps entre des occurences successives de même événements
- La gigue la plus importante à observer sera la gigue de release, qui correspond à la variation des temps de démarrage de tâches périodiques

Gigue

La gigue observée peut l'être sur différentes variables du système. Nous pouvons définir les gigues suivantes:

- **La gigue de release relative** (Relative Release Jitter) d'une tâche est la déviation maximale des temps de démarrage de deux instances consécutives:

$$RRJ_i = \max_j |(s_{i,j} - r_{i,j}) - (s_{i,j-1} - r_{i,j-1})| \quad (1)$$

- **La gigue de release absolue** (Absolute Release Jitter) d'une tâche est la déviation maximale des temps de départ sur toutes les instances:

$$ARJ_i = \max_j (s_{i,j} - r_{i,j}) - \min_j (s_{i,j} - r_{i,j}) \quad (2)$$

Gigue

- **La gigue de fin relative** (Relative Finishing Jitter) d'une tâche est la déviation maximale des temps de fin de deux instances consécutives:

$$RFJ_i = \max_j |(f_{i,j} - r_{i,j}) - (f_{i,j-1} - r_{i,j-1})| \quad (3)$$

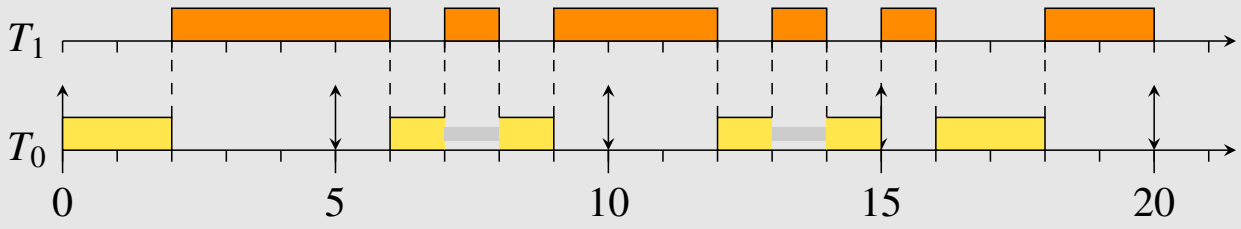
- **La gigue de fin absolue** (Absolute Finishing Jitter) d'une tâche est la déviation maximale des temps de fin sur toutes les instances:

$$AFJ_i = \max_j (f_{i,j} - r_{i,j}) - \min_j (f_{i,j} - r_{i,j}) \quad (4)$$

Gigue: Exemple

- A partir de cet ordonnancement de tâches:

2 tâches



- Déterminer les valeurs de gigue pour la tâches T_0:

Type	T_0
RRJ_i	1
ARJ_i	2
RFJ_i	2
AFJ_i	3