

Méthodes d'ordonnancement (I)

Hongyang QU

Ordonnements classiques

- PAPS (FIFO : First In First Out) : premier arrivé, premier servi
- Tourniquet (round robin) : à tour de rôle
- l'ordonnement par priorité

Les décisions d'ordonnement

- Ordonnement : choisir la tâche qui devient la tâche actuelle.
 - En ligne (“on line”)
 - Hors ligne (“off line”)

Les types d'algorithmes d'ordonnancement

- Statique

L'ordonnancement est déterminé préalablement à l'exécution.

- Dynamique

L'ordonnancement est déterminé lors de l'exécution.

Deux concepts

- L'ensemble des tâches est “**faisable**” s'il existe un ordonnancement de ces tâches qui respecte les contraintes temporelles associées à ces tâches.
- Un ordonnanceur “**optimal**” est un ordonnanceur qui peut produire un ordonnancement pour tout ensemble faisable de tâches.

Les catégories d'algorithmes d'ordonnancement

1. Algorithmes statiques pilotés par table
2. Algorithmes statiques préemptifs basés sur les priorités
3. Algorithmes dynamiques basés sur une planification à l'exécution
4. Algorithmes dynamique basés sur la notion de meilleur effort (« best effort »)

Les caractéristiques de tâches

- Tâches périodiques
- Tâches sporadiques
- Tâches aperiodiques

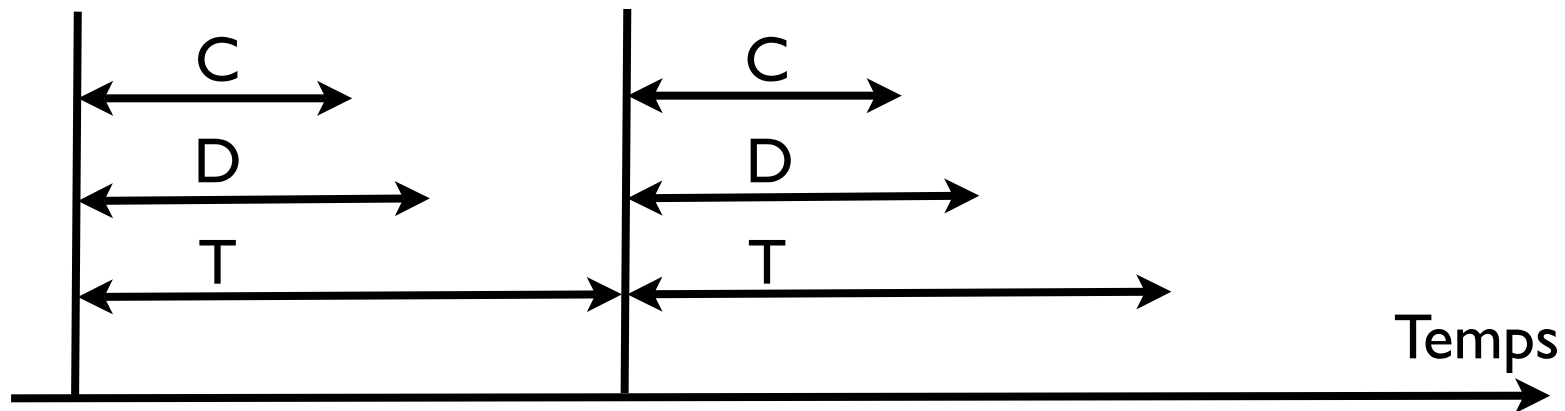
Des tâches périodiques

- Elles sont activées à intervalles réguliers

C : capacité = temps d'exécution dans le pire des cas

D : échéance

T : période d'activation de la tâche



Des tâches non périodiques

- Tâches sporadiques

- activées à des instants irréguliers
- un temps minimal entre deux activation d'instances successives
- échéances fortes

- Tâches apériodiques

- activées à des instants irréguliers
- Pas de temps minimal entre deux activations d'instances successives
- échéances lâches

Des tâches dépendantes et indépendantes

- indépendante : les tâches ne partagent aucune ressource commune
- dépendante : les tâches partagent des ressources communes

Un modèle d'ordonnancement cyclique

- une séquence est une procédure définie par l'utilisateur;
- un processus est une suite ordonnée de séquences;
- l'ordonnancement des processus est régi par un calendrier.

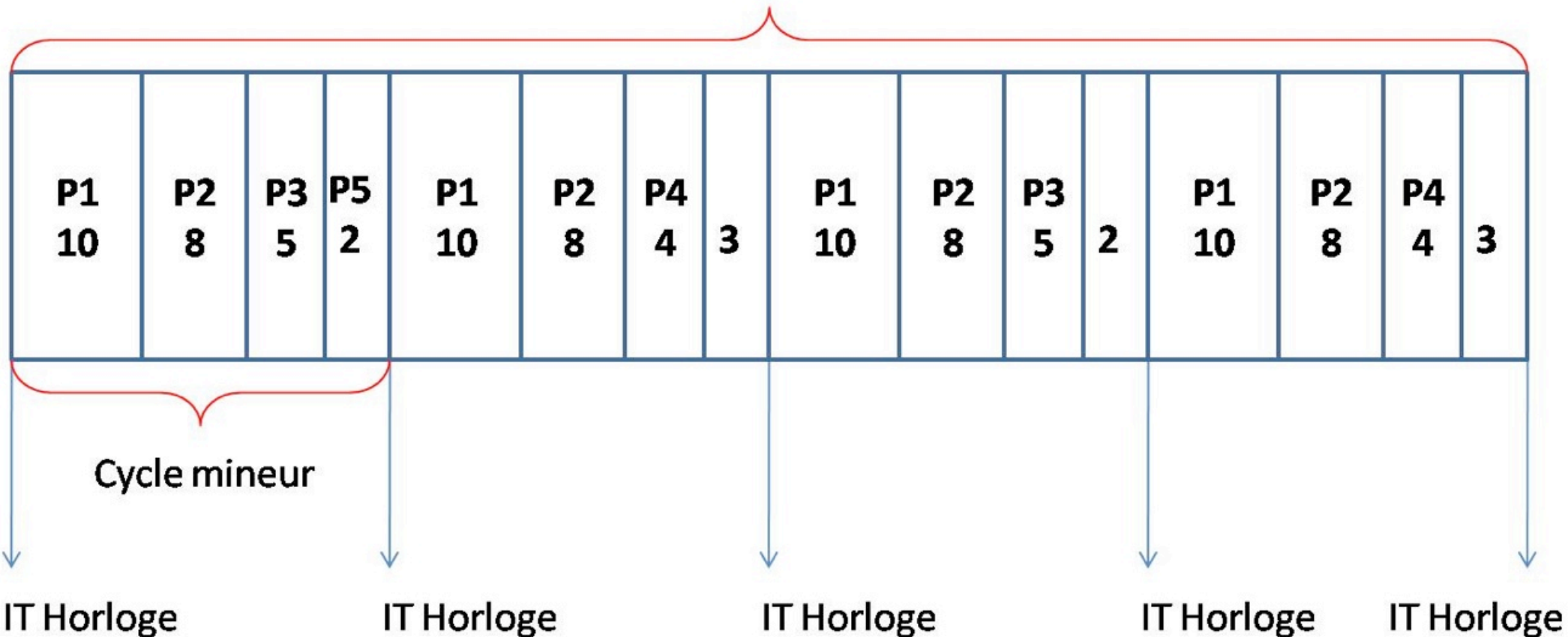
Exemple (I)

- $P1 : T1 = 25, C1 = 10, D1 = 25$
- $P2 : T2 = 25, C2 = 8, D2 = 25$
- $P3 : T3 = 50, C3 = 5, D3 = 50$
- $P4 : T4 = 50, C4 = 4, D4 = 50$
- $P5 : T5 = 100, C5 = 2, D5 = 100$

Exemple (2)

cycle majeur : les processus cycliques (PPCM des "T"s)
cycle mineur : le rythme des interruptions d'horloge

Cycle majeur

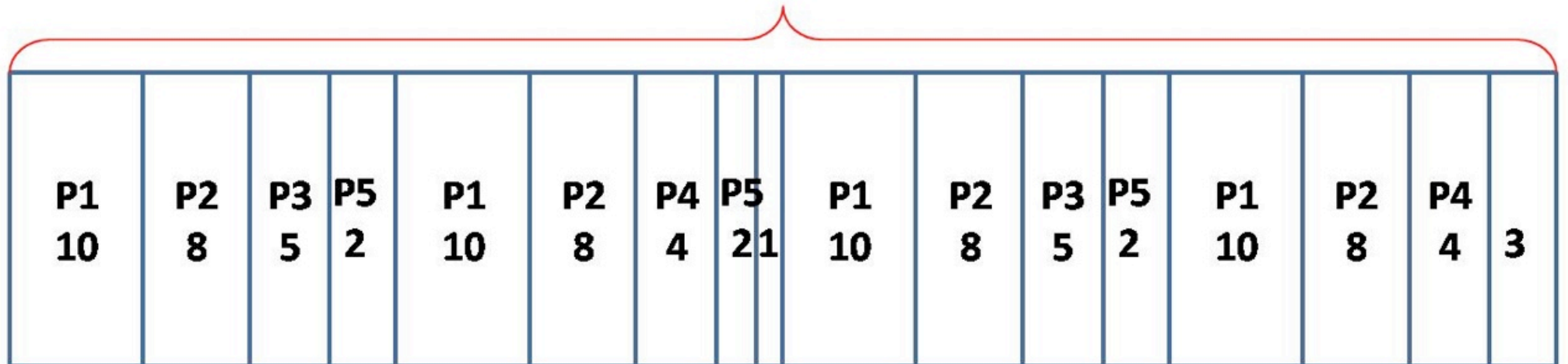


Exemple (3)

- $P1 : T1 = 25, C1 = 10, D1 = 25$
- $P2 : T2 = 25, C2 = 8, D2 = 25$
- $P3 : T3 = 50, C3 = 5, D3 = 50$
- $P4 : T4 = 50, C4 = 4, D4 = 50$
- $P5 : T5 = 100, C5 = 6, D5 = 100$

Exemple (4)

Cycle majeur



Cycle mineur

IT Horloge

IT Horloge

IT Horloge

IT Horloge

IT Horloge

Des critères utilisés par des ordonnancements en ligne

- priorité (“priority”)
- échéance (“deadline”)
- marge (“laxity”)
 - $\text{marge} = \text{échéance} - \text{durée restante de temps de traitement}$

Ordonnancement en ligne

- HPF (“Highest Priority First”)
priorité la plus élevée d’abord
- EDF (“Earliest Deadline First”)
échéance la plus proche d’abord
- LLF (“Least Laxity First”)
marge la plus petite d’abord

Ordonnancement RM

- RM (“Rate Monotonic”) : Monotone par taux
- Critères:
 - Les tâches sont périodiques et sont à l'état PRET au début de chaque période ; leur échéance se situe à la fin de la période ; elles ne se suspendent pas elles-mêmes en cours d'exécution.
 - Les tâches peuvent être préemptées, mais on néglige le temps de commutation et d'ordonnancement ;
 - Les tâches sont indépendantes les unes des autres (pas de synchronisation entre tâches) ;
 - Le temps d'exécution dans le pire des cas (appelé aussi capacité) de chacune des tâches est connu.

attribution des priorités aux tâches

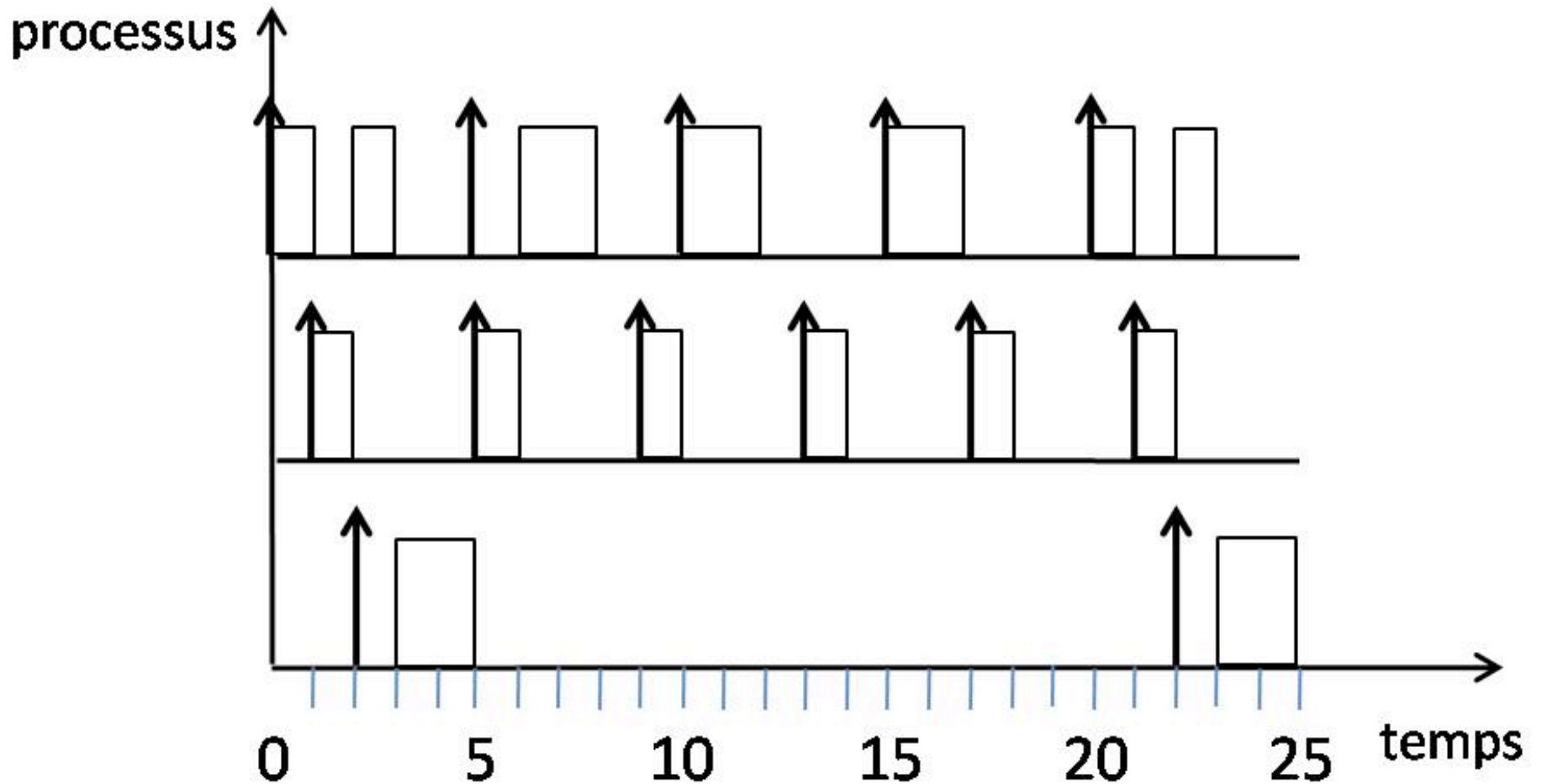
- T_i : la période de la tâche i

- la priorité de tâche i : $\frac{1}{T_i}$

Exemple (I)

- ◆ S_i : le temps d'arrivé de tâche i
- P1 : $S1 = 0, C1 = 2, T1 = D1 = 5$
- P2 : $S2 = 1, C2 = 1, T2 = D2 = 4$
- P3 : $S3 = 2, C3 = 2, T3 = D3 = 20$

Exemple (2)



Condition suffisante de l'analyse RM (I)

- Le taux d'utilisation du processeur :

$$U = \sum_{i=1}^n \frac{\textit{durée}_i}{\textit{période}_i}$$

Condition suffisante de l'analyse RM (2)

- C_i : le temps d'exécution de la tâche i ,
- T_i : la période de la tâche i ,
- D_i : la échéance de la tâche i ($D_i = T_i$).

$$\forall i, 1 \leq i \leq n, U_i = \sum_{j=1}^i \frac{C_j}{T_j} \leq i(2^{\frac{1}{i}} - 1)$$

Condition suffisante de l'analyse RM (3)

- condition suffisante mais pas nécessaire

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

pour $n=1$, $U=1$,

$n=2$, $U=0.828$,

$n>10$, $U=\ln(2)=0.69$

Principes d'obtention de la condition suffisante

- Considérez seulement deux tâches (T1, C1, T2, C2)

$$U = C1/T1 + C2/T2$$

Deux opérateurs

- $\lceil \rceil$: plafond (“ceiling”)

$$\lceil x \rceil = x \text{ si } x \text{ est un nombre entier;} \\ a + 1 \text{ si } x = a.b \text{ (} b > 0 \text{)}$$

- $\lfloor \rfloor$: plancher (“floor”)

$$\lfloor x \rfloor = x \text{ si } x \text{ est un nombre entier;} \\ a \text{ si } x = a.b \text{ (} b > 0 \text{)}$$