Université Djilali BOUNAAMA, Khemis Miliana
مليانة خميس بونعامة جيلالي جامعة
Faculté des Sciences et de la Technologie
والتكنولوجيا العلوم كلية

Série d'exercices 6: Partie B : Introduction au temps réel

Exercice 1 : Classification de contraintes temporelles

Pour chacun des systèmes suivants, dites s'il s'agit d'un système temps réel « dur » (hard), « mou » (soft) ou « ferme » (firm, notion intermédiaire où une échéance dépassée rend la donnée inutile mais sans catastrophe). Justifiez.

1. Système de freinage ABS d'une voiture.
2. Lecteur DVD qui doit décoder une trame vidéo toutes les 40 ms.
3. Serveur de transactions bancaires en ligne (délai maximal de réponse de 2 secondes, au-delà l'utilisateur peut recommencer).
4. Commande d'un bras robotique en soudage de précision.

Exercice 2 : Calcul de latence et de gigue

Énoncé :
Une tâche T est exécutée périodiquement sur un microcontrôleur. On mesure les instants de déclenchement (release time) et de fin réelle (completion time) sur 4 cycles :

Cycle	Instant déclenchement théorique	Instant fin réel
1	0 ms	5 ms
2	10 ms	17 ms
3	20 ms	24 ms
4	30 ms	34 ms

1. Calculez la latence (response time) pour chaque cycle.
2. Calculez la gigue d'activation (release jitter) et la gigue d'achèvement (completion jitter) sur ces 4 mesures.
3. Le système est-il déterministe ? Pourquoi ?

Exercice 3 : Ordonnancement et échéance

Deux tâches périodiques temps réel **dures** :

- Tâche A : période = 10 ms, durée d'exécution (WCET) = 4 ms, échéance = période.
- Tâche B : période = 15 ms, durée = 6 ms, échéance = période.

1. Vérifiez la faisabilité avec l'algorithme Rate Monotonic (utilisez le critère de Liu & Layland pour deux tâches).
2. Si on ajoute une tâche C : période = 30 ms, durée = 12 ms, le système reste-t-il ordonnançable ?

Exercice 4 : Choix d'architecture logicielle

On conçoit un drone autonome avec :

- une boucle de contrôle de vol (échéance 1 ms, critique)
- une transmission vidéo (échéance 30 ms, perte de trames acceptable)
- une journalisation des données sur carte SD (pas d'échéance stricte).

Proposez une architecture logicielle adaptée (monolithique super-loop, noyau temps réel préemptif, ou système d'exploitation généraliste avec priorités) en justifiant.

Exercice 5: Déterminisme vs performance

Un système embarqué doit acquérir un capteur toutes les 5 ms. Deux solutions :
A) Interruption matérielle + routine courte dans l'ISR, puis traitement différé.
B) Thread temps réel avec usleep(5 ms) sous un OS standard.

Expliquez pourquoi la solution A est plus déterministe. Calculez la gigue possible pour B sachant que l'OS peut avoir une latence de réveil jusqu'à 2 ms.

Exercice 6: Identifier les métriques sur un chronogramme

On donne le chronogramme suivant d'une tâche périodique temps réel (échéance = période = 20 ms) :

t (ms)	0	5	10	15	20	25	30	35	40	45
Événement	Arrivée	Début exéc.	Fin exéc.	Arrivée	Début	Fin	Arrivée	Début	Fin	Arrivée

(On suppose que la durée d'exécution est de 5 ms à chaque fois, mais l'arrivée suivante peut survenir avant la fin de la précédente : c'est de la libération périodique sans attente.)

Calculez :

1. La latence de chaque instance.
2. La gigue d'achèvement.
3. L'instance manque-t-elle son échéance ? Pourquoi ?

Université Djilali BOUNAAMA, Khemis Miliana
مليانة خميس بونعامة جيلالي جامعة
Faculté des Sciences et de la Technologie
والتكنولوجيا العلوم كلية

Solution Série 6

Exercice 1 :

1. **Dur** – un dépassement d'échéance peut entraîner un accident grave.
2. **Ferme** – une trame arrivée trop tard sera ignorée (image sautée), mais le système continue sans dommage critique.
3. **Mou** – un retard dégrade l'expérience utilisateur mais n'est pas catastrophique.
4. **Dur** – une commande en retard peut provoquer une collision ou une soudure hors tolérance.

Exercice 2 :

1. Latence = fin réelle – déclenchement théorique :
   Cycle 1 : 5 ms
   Cycle 2 : 7 ms
   Cycle 3 : 4 ms
   Cycle 4 : 4 ms
2. **Gigue d'activation** = écart entre intervalles réels d'activation et intervalle théorique (10 ms).
   Intervalles réels : 10 ms (t2-t1) ? Non : déclenchement réel ? On suppose ici que déclenchement théorique = réel (pas de jitter d'activation donné). Donc jitter d'activation = 0.
   Gigue d'achèvement = variation des latences : min=4, max=7, gigue = 7-4 = 3 ms.
3. Le système n'est pas complètement déterministe car les latences varient (3 ms de gigue). Un système déterministe aurait une latence constante.

Exercice 3 :

1. Utilisation U = 4/10 + 6/15 = 0,4 + 0,4 = 0,8.
   Pour deux tâches, limite RM = 2×(2^(1/2)-1) ≈ 0,828.
   0,8 ≤ 0,828 → ordonnançable.
2. Nouvelle utilisation U = 0,8 + 12/30 = 0,8 + 0,4 = 1,2 > 1 → Impossible, même avec un ordonnanceur optimal. Donc non ordonnançable.

Exercice 4

Architecture conseillée : **noyau temps réel préemptif** (ex : FreeRTOS, VxWorks) avec priorités fixes.

- Boucle de contrôle : priorité la plus haute, pour respecter l'échéance dure.
- Vidéo : priorité moyenne, temps réel mou.

- Journalisation : priorité basse, tolère les retards.

Un super-loop ne permettrait pas de garantir l'échéance de 1 ms en présence d'une vidéo.
Un OS généraliste (Linux sans RT patch) peut introduire trop de gigue.

Exercice 5

- Solution A : l'ISR répond à l'événement matériel avec une latence très faible (quelques cycles CPU). Le déclenchement est précis et déterministe.
- Solution B : usleep(5) + latence de réveil (due à l'ordonnanceur) peut donner un intervalle réel entre 5 ms et 7 ms.
  Gigue possible = 2 ms (si aucun autre facteur).

Donc A est plus déterministe car le temps de réponse est quasi constant et non soumis à l'ordonnanceur.

Exercice 6

D'après le tableau :

- Instance 1 : arrivée à 0, fin à 5 → latence 5 ms.
- Instance 2 : arrivée à 20, fin à 25 → latence 5 ms.
- Instance 3 : arrivée à 40, fin à 45 → latence 5 ms.

Mais attention : dans votre tableau, les arrivées sont à 0, 20, 40 (période 20 ms) mais les débuts sont à 5, 25, 45 ? Non, revérifions :

Vous avez écrit : t=0 arrivée, t=5 début, t=10 fin → latence = 10-0 = 10 ms.
t=15 arrivée, t=20 début, t=25 fin → latence = 25-15 = 10 ms.
t=30 arrivée, t=35 début, t=40 fin → latence = 40-30 = 10 ms.

Donc latence constante = 10 ms.
Gigue d'achèvement = 0 ms (déterministe).

Échéance = 20 ms après arrivée :
Instance 1 : échéance à 20, fin à 10 → respectée.
Instance 2 : échéance à 35, fin à 25 → respectée.
Instance 3 : échéance à 50, fin à 40 → respectée.

Aucune échéance manquée car exécution courte (5 ms) et période 20 ms, avec latence 10 ms.