

# Examen Temps Réel -

Enseignant, auteur du sujet : Emilie Poisson Caillault

17/03/2025

## Consignes :

- Durée examen : 120 minutes
- Aucun document, aucun matériel ou montre numérique
- Aucune calculatrice autorisée, vous approcherez vos calculs à un chiffre après la virgule si besoin.
- 4 points pour chaque exercice.

## Rappels :

- Critère de Liu Layland :
  - ordonnancement statique :  $Uo \leq n(2^{1/n} - 1) = 0.78$  pour  $n=3$  et  $0.75$  pour  $n=4$
  - ordonnancement dynamique :  $Uo \leq 1$
- Théorème de la zone critique :  $W_i(t) = C_i + \sum_{j|priorité(T_j) > priorité(T_i)} C_j \times \left\lceil \frac{t}{P_j} \right\rceil$

## Exercice A - Application Taches périodiques indépendantes

Un Système de Surveillance et de Contrôle d'un Bâtiment Intelligent est composé de trois tâches principales à réveil simultané ( $r=0$ ):

- La tâche T1 de Contrôle de la température intérieure vérifie les capteurs de température dans différentes zones du bâtiment et ajuste le système de chauffage ou de climatisation si nécessaire. Cette tâche à échéance sur requête a une durée d'exécution de 4 seconds et une périodicité de 20 secondes.
- Une tâche T2 de Surveillance des caméras de sécurité Analyse les flux vidéo des caméras de sécurité pour détecter des mouvements ou des comportements suspects. Cette tâche à échéance sur requête a une durée d'exécution de 2 secondes et une périodicité de 10 secondes.
- Une tâche T3 de gestion de requêtes, telles que les demandes d'accès ou les alertes d'urgence. Cette tâche à échéance sur requête a une durée d'exécution de 2 secondes et une périodicité de 5 secondes.

Proposer une solution déterministe ordonnançable. Justifier votre choix.

Tracer le graphe d'activités des tâches sur l'hyperpériode.

Expliciter les 5 premiers appels à l'ordonnanceur.

## Exercice B - Application Taches apériodiques

Considérer maintenant que les requêtes sont apériodiques.

Soient les deux requêtes suivantes :

- La première requête (notée A1) concerne la demande d'accès à une zone restreinte. Elle vérifie les autorisations et ouvre la porte d'accès si les conditions sont remplies. Elle arrive : 3 secondes après le démarrage du système et a une durée d'exécution de 2 secondes.
- La seconde (notée A2) concerne une alerte d'incendie déclenchée 6 secondes après le démarrage du système avec une durée d'exécution de 4 secondes.



T1 et T2 s'exécutent périodiquement pour assurer le confort et la sécurité du bâtiment. T3 agit comme un serveur, vérifiant périodiquement s'il y a des tâches aperiodiques à traiter.

A1 et A2 sont des événements imprévus qui doivent être traités rapidement par T3 sans perturber les tâches périodiques critiques.

Proposer une solution satisfaisante (type du serveur). Justifier

Expliciter l'ensemble des structures utilisées.

Donner les temps de réponse de chaque tâche périodique et aperiodique.

### Exercice C - Tâches synchronisées

Soit 4 tâches dont les paramètres sont les suivants : A(0,2,5) ; C(1,1,20) ; B(4,2,10) ; D(0,1,20); E(1,3,20). Cette configuration est répétée tous les 40 unités de temps avec les contraintes suivantes : A et B doit être réalisée avant C; C avant D.

Expliciter toutes les solutions possibles pour ordonnancer cette configuration de tâche avec précedence. Détailler.

### Exercice D - Programmation Tâches synchronisées

Soit le code suivant.

```
import random, time
from threading import Event, Thread

def do1(lettre,nfois,evt):
    for i in range(nfois):
        print(lettre)
        evt.set();

def do2(lettre,nfois,evt):
    print(lettre)
    evt.wait();
    evt.clear();
    for i in range(nfois):
        print(lettre)

if __name__ == '__main__':
    event = Event()
    thread3 = Thread(target=do2,args=("C",2,event))
    thread2 = Thread(target=do2,args=("B",5,event))
    thread1 = Thread(target=do1,args=("A",2,event))

    thread3.start()
    thread1.start()
    thread2.start()

    thread1.join()
    thread2.join()
    thread3.join()

    print("\n Fin Main\n.")
```

Donner et expliquer deux traces (logs) possibles d'exécution en détaillant les structures actives, prêtes et bloquées.



## Exercice E - Questions de Cours

Vous vous placez dans la configuration suivante à quatre tâches telles que :

- T1 travaille sur deux ressources R1 et R2 ainsi : R1 R1 (R1R2) R2
- T2 travaille sur deux ressources R2 et R3 ainsi : R2 R2 R3 R3 R3
- T3 travaille sur deux ressources R3 et R2 ainsi : R3 R3 (R2R3) R2
- T4 travaille sans aucune ressource : EE

Un protocole de partage de ressources est-il indispensable ici ? Citer les problèmes possibles.

Quel protocole vous semble le plus approprié ? Détailler les structures et primitives nécessaires à ce protocole choisi. Expliciter les algorithmes d'entrée et sortie en section critique.

A quoi sert l'indivisibilité des primitives ?