Travaux Dirigés n°3
Java Temps Réel

Synchronisations & Protocoles d'accès

Le but de ce td est d'étudier le comportement d'un système de tâches en cas de partage de ressource, de comprendre et de mettre en évidence le phénomène d'inversion de priorité non bornée ainsi que le protocole d'accès à héritage de priorité.

▸ **Exercice 1.** *Inversion de priorité non bornée*

Vous allez mettre en oeuvre le système $S_\tau 3$ et mettre en évidence le phénomène d'inversion de priorité non-borné. Dans ce cas de figure la synchronisation entre les tâches τ_1 et τ_3 se fera a l'aide d'un objet ressource.

Tâche	Start Time (sec)	Coût	Echéance	Période	Priorité
τ_1	3	$1 + 5 + 1$	20	20	14
τ_2	5	3	20	20	13
τ_3	1	$1 + 5 + 1$	20	20	12

TABLE 1 – système $S_\tau 3$

1. **Construction du système** $S_\tau 3$

Vous allez écrire les classes `TD3`, `Ressource1`, `CustomRealtimeThread`, `CustomRealtimeThread2`.

- *La classe* `Ressource1`

Cette classe offre la méthode `public static synchronized void get()` *qui :*
– *affiche le type de moniteur utilisé (protocole) ;*
– *affiche "entrée dans la ressource 1" ;*
– *exécute une boucle de 5 secondes (utilisation de la ressource) avec un affichage à chaque seconde ;*

– *affiche "sortie de la ressource 1" ;*

- *La classe* `CustomRealtimeThread`

  Elle va servir à l'exécution des threads τ_3 et τ_1. Les threads sont périodiques, on n'exécutera que leurs deux premières périodes.
  La méthode `run()` *:*
  – *exécute 1s normale ; affichage*
  – *utilise la ressource 1 pendant 5s (appel de la méthode* `Ressource1.get()`*)*
  – *exécute 1s normale ; affichage*

- *la classe* `CustomRealtimeThread2`

  Elle va servir à l'exécution du thread τ_2. Le thread est périodique, on n'exécutera que ses deux premières périodes. Sa méthode `run()` *s'exécute 3s ; affichage à chaque seconde.*

- *la classe* `TD3`

  Elle étend `RealtimeThread` *:*
  * *Sa méthode* `main()` *:*
    – *crée un objet td3 de type TD3 en lui donnant une priorité de 20 (le plus prioritaire).*
    – *lance le thread td3*
  * *Sa méthode* `run()` *:*
    – *affiche le type de moniteur de synchronisation par défaut(*`MonitorControl.getMonitorControl()`*)*
    – *crée t3 instance de* `CustomRealtimeThread` *(voir les paramètres dans la table 1)*
    – *crée t1 instance de* `CustomRealtimeThread` *(voir les paramètres dans la table 1)*
    – *crée t2 instance de* `CustomRealtimeThread2` *(voir les paramètres dans la table 1)*
    – *lance t3,t1,t2*

2. **Prévision du comportement**
   *Tracez un diagramme d'exécution du système $S_\tau 3$ dans l'hypothèse d'un protocole a inversion de priorité non bornée puis dans l'hypothèse d'un protocole d'accès à héritage de priorité (*PIP*). Pour cela, reportez vous au tableau 1 et a la description des classes* `CustomRealtimeThread*.`

3. **Execution du système $S_\tau 3$**

Lancez $S_\tau 3$ sur la plateforme RI et stockez les résultats dans un fichier. Comparez les résultats obtenus à ceux tracés sur le diagramme temporel.

▸ **Exercice 2.** *Héritage de priorité Transitif*

Dans cet exercice vous allez définir (voir le cours) le scénario qui montrera si l'héritage est bien transitif.

▸ **Exercice 3.** *PIP/PCP/PCE - Simulation Cheddar*

1. *Installation de Cheddar http ://beru.univ-brest.fr/ singhoff/cheddar/index.html*
2. *Simulez le système avec Cheddar en modifiant les offsets des tâches τ_1 et τ_2. Offset de $\tau_1 = 4$; offset de $\tau_2 = 3$.*
3. *Créez le système de tâches décrit dans la table 2. Crééz les deux ressources (Bleue, Jaune) et définissez leur utilisation. Par exemple τ_1 utilise Bleue (begin=2 ; end=4).*
4. *Simuler le système suivant en utilisant PIP, PCP, PCE*

Tâche	Start Time (sec)	Coût	Echéance	Période	Priorité
τ_1	5	8 : 1+3(Bleue)+1+2(Jaune)+1	30	30	4
τ_2	2	5 : 1+3(Jaune)+1	30	30	2
τ_3	0	6 : 1+3(Bleue)+1	30	30	1

TABLE 2 – système Sτ4

Tâche	Begin	End
τ_1	2	4
τ_3	2	4

TABLE 3 – Paramètres de la Ressource Bleue

Tâche	Begin	End
τ_1	6	7
τ_2	2	4

TABLE 4 – Paramètres de la Ressource Jaune

▸ **Exercice 4.** *Comparaison PCP/PCE - Simulation Cheddar*

1. *Ajoutez la tâche τ_0.*
2. *Montrez sur la simulation les avantages et les inconvénients de chaque protocole de synchronisation. Pour cela on observera le nombre de changement de contexte, le nombre de préemptions et les temps de réponse des tâches.*

Tâche	Start Time (sec)	Coût	Echéance	Période	Priorité
τ_0	1	1	10	10	3

TABLE 5 – système Sτ4