

TP –Temps-réel

Exercice 1 : étude de la faisabilité

Nous allons étudier la faisabilité d'un ensemble de tâches à travers l'exemple suivant :

Tâche	Durée d'exécution	Echéance	Période	priorité
Thread 1	2	7	7	20
Thread 2	3	11	11	15
Thread 3	5	13	13	10

Toutes les valeurs sont exprimées en seconde.

- 1) Développer une application afin de vérifier la faisabilité de l'ensemble de tâches en utilisant un ordonnanceur statique (HPF : Highest Priority First).
- 2) Généraliser votre application pour un nombre quelconque de tâches temps réelles.
- 3) Généraliser votre solution en intégrant d'autres ordonnanceurs statiques (ex. RM, DM).
- 4) [Optionnelle] Intégrer le cas d'un ordonnanceur préemptif / non préemptif.

Exercice 2 : Ordonnanceur EDF

Earliest deadline first (EDF) est un algorithme d'ordonnancement dynamique le plus optimal utilisé dans les systèmes d'exploitation en temps réel. A chaque fois qu'un événement d'ordonnancement se produit (fin d'une tâche, lancement d'une nouvelle tâche, etc.), la file d'attente est parcourue à la recherche du processus le plus proche de son échéance absolue.

- 1) Développer un simulateur proposant EDF comme ordonnanceur.
- 2) Vérifier votre application en traçant l'exécution d'un exemple de tâches (messages sur console ou interface graphique).

Exercice 3 : mini projet

L'objectif de cet exercice est de préparer un environnement embarqué doté d'un système d'exploitation temps réel.

Pour la première version, vous avez la possibilité de choisir entre 2 solutions :

- 1) Installation d'un noyau Realtime-Pi sur une raspberry Pi [1].
- 2) Installation d'un noyau NiRTOS sur une arduino [2].

Il faut prévoir des tests de conformité avec le temps réel (ex. RT-tests [3, 4]).

Il est possible de faire une 2^{ème} version. Elle se résume à l'utilisation d'un noyau un peu plus élaboré (ex. FreeRTOS [5]).

Référence :

1. <https://github.com/guysoft/RealtimePi>
2. <https://github.com/greiman/NiRTOS-Arduino>
3. https://linuxsoft.cern.ch/cern/centos/7/rt/x86_64/repoview/rt-tests.html
4. <https://github.com/jlelli/rt-tests>
5. <https://www.freertos.org/>