

TD2 d'ordonnancement temps réel

Capteurs/Actionneurs – Polytech' Nice

Exercice 3 – Comparaison d'ordonnancements

Soient deux tâches T1 et T2 définies par les paramètres suivants : $S = 0$, $P_1 = 8$, $P_2 = 10$, $C_1 = 4$ et $C_2 = 5$. Les délais critiques sont égaux aux périodes (soient $\forall i : D_i = P_i$).

- Dessinez sur les 24 premières unités de temps l'ordonnancement généré par EDF (en mode préemptif). Existe-il des échéances manquées ?
- Dessinez sur les 24 premières unités de temps l'ordonnancement généré par un algorithme à priorités fixes RM (en mode préemptif). Existe-il des échéances manquées ?
- Concluez

Exercice 4 – Politique LLF

Soient trois tâches T1(1,8), T2(2,4) et T3(4,10). Les délais critiques sont égaux aux périodes (soient $\forall i : D_i = P_i$).

- Dessinez sur les 24 premières unités de temps l'ordonnancement généré par RMS (en mode préemptif). Existe-il des échéances manquées ?
- Dessinez sur les 24 premières unités de temps l'ordonnancement généré par un algorithme à LLF (en mode préemptif). Existe-il des échéances manquées ?
- Concluez

Exercice 5 - Partage de ressources

Dans cet exercice, on souhaite montrer l'impact d'une inversion de priorité sur l'ordonnancement d'un jeu de tâches. Soient trois tâches définies par les paramètres suivants : $S_1 = S_2 = S_3 = 0$, $P_1 = 6, C_1 = 2$, $P_2 = 8, C_2 = 2$, $P_3 = 12$ et $C_3 = 5$. Les délais critiques sont égaux aux périodes (soient $D_i = P_i$). On utilise un algorithme à priorité fixe RM pour ordonnancer les tâches (mode préemptif). Les tâches T1 et T3 se partagent une ressource qu'elles accèdent en exclusion mutuelle. T1 accède à la ressource durant la deuxième unité de temps de sa capacité. T3 accède à la ressource durant la totalité de sa capacité.

- Dessinez sur la période d'étude l'ordonnancement généré par un algorithme à priorité fixe RM. Vous indiquerez les moments d'accès exclusif à la ressource ainsi que le moment où l'inversion de priorité intervient
- On suppose que l'on utilise la méthode d'héritage simple : une tâche qui bloque une autre plus prioritaire qu'elle, exécute la section critique avec la priorité de la tâche bloquée. Cette méthode d'héritage s'appelle aussi PIP pour Priority Inheritance Protocol. Redessinez l'ordonnancement sur la période d'étude et indiquez le moment sur le graphe où l'inversion de priorité est évitée.