

Ordonnancement temps réel contraint par l'énergie.

Ce projet consiste à étudier l'ordonnancement temps réel contraint par l'énergie. Il s'agit d'exposer les méthodes à mettre en œuvre afin d'optimiser les capacités des systèmes embarqués tout en réduisant au maximum la consommation énergétique. Celles-ci peuvent être matérielles ou logicielles. Dans le cadre de ce projet, nous focaliserons notre étude sur les solutions logicielles.

Dans un premier temps, nous commençons par définir ce que sont les systèmes embarqués. Les systèmes embarqués réalisent des tâches généralement répétitives et connues d'avance. Optimiser leurs capacités va revenir à faire en sorte qu'ils puissent effectuer toutes les opérations qui leur sont dévolues à temps. La solution la plus simple serait d'augmenter la vitesse de calcul du processeur, mais cette solution augmente considérablement l'énergie électrique consommée. On comprend ici que les deux buts de ce travail, à savoir optimiser à la fois les facultés de calcul et la consommation d'énergie, risquent d'entrer en conflit.

C'est ainsi que nous envisageons une solution alternative : l'ordonnancement. Choisir un ordre optimal pour les différentes tâches à effectuer permet de réaliser plus d'opérations dans un temps égal et avant l'échéance fixée pour chaque travail.

Différents types d'algorithmes d'ordonnancement permettent de satisfaire ces contraintes.

Le premier que nous étudions, l'algorithme *RMS*, est un algorithme statique. Cela signifie qu'il fixe à l'avance les priorités de chaque tâche et ne les modifie plus pendant le fonctionnement du système.

Le second, l'algorithme *EDF*, est un algorithme dynamique. Il va donc changer l'ordre de ses priorités selon la situation. Son critère est le suivant : le travail le plus urgent a la priorité la plus grande.

On peut penser que *EDF* est forcément plus efficace, ce qui est évidemment exact, mais le nombre de calculs mis en œuvre pour « recalculer » l'ordre de ses priorités est lui aussi une source de travail et de consommation d'énergie, c'est pourquoi il n'est par exemple pas indiqué pour les systèmes très faibles en termes de capacités de calcul et d'énergie.

Enfin, nous présentons également une méthode d'économie d'énergie, applicable aux algorithmes décrits plus haut : Le *Reclaiming*. Celle-ci consiste à réduire la vitesse de calcul lorsqu'il y a des périodes d'inactivité du processeur de sorte que celui-ci ne consomme pas d'énergie inutile en fonctionnant à une cadence trop élevée pour rien. Tout cela bien sûr en continuant de s'assurer d'accomplir toutes les tâches dans les temps impartis.

Nous avons également implémenté un logiciel permettant de réaliser des simulations faisant intervenir les deux algorithmes décrits plus haut, ceci afin de nous permettre de visualiser les exécutions des différentes tâches au cours du temps, et d'afficher notamment les temps d'oisiveté durant lesquels aucune tâche ne se présente, etc.