

1 Laboratoire

Retour sur les different wait possibles (*osThread*, *Thread::wait*, *wait_ms*). Le premier numero 1 est un deadlock. Le numero 2 sur les mutex et la deifferences des delais. Refaire le Laboratoire

2 RTOS

La difference en hard real-time et soft real-time est que:

- Le hard real-time doit absolument atteindre les bon deadlines au bon moment dans le temps. On parle d'evenement critique. Ex: Un pace-maker.
- Le soft n'est pas super constrain dans le temps. On parle d'evenement non-critique. un live stream qui perd un packet.

Nous allons analyser deux type de OS, soit RTOS(Real-Time OS) et GPOS(System general d'exploitation). Un GPOS est comme Windows ou Linux. tandis qu'un RTOS est simple et n'installe pas aucun driver qui ne sont pas utiliser. RTOS est plus performant, mais il ne fait qu'une seule chose a la fois. Le RTOS est habituellement designer pour s'executer de la ROM/RAM directment sans utiliser de stockage externe. De plus, le RTOS est plus facile de changer de plateforme. Le GPOS est nettement plus gros que les RTOS. Ils sont assez gros pour ne pas etre possible de les mettres sur les mbed.

La fiabilite des RTOS on une unite de mesure precise. On parle des 9. si on parle de 99.9%, on considere que sur 1 ans, on ne pourras pas utiliser la machine durant 9h. On y retrouve les Desktop dans cette categorie. Quand on parle de 99.99% on parle de 1h. On l'y retrouve les Serveurs ou les entreprises. Quand on parle de 99.999% on parle de 5 min. On l'y retrouve encore des serveur mais de type de carriere class. Quand on parle de 9.9999%, on parle de 31 seconde par annee. On l'y retrouve les carriers switchs. Cette categorie est extremement rare.

Un autre point sur les RTOS, on considere leur predictibilite. On parle de connaitres leur etat dans plusieurs annee. On doit connaitre leur performance au fils des annees. RTOS est souvent modifier au fil des annees. On demarre le RTOS apres le boot loader, comme montre dans le diagramme ci dessus.

Reset Vector
BootLoader
HW
RTOS Init
RTOS composant
RTOS Start
Program

3 Multi-Tache

On parle de trois etats possible d'une tache, READY, RUNNING et BLOCKED. Ready est quand la tache est pret a etre executer. Running est quand elle est ex-ecuter. Blocked est quand elle est blocker. La prioriasation des taches permet la transision entre Ready et Running pour certaines taches.Cela peut arriver qu'une tache soit redui de priorite et retourne en mode READY. Voir Schema ci-dessus. (A TROUVER DANS LE LIVRE).

Dans RTOS, on a plusieurs methode de priorisation des taches, Soit round-robin, Par priorisation vonlontaire (preemptive scheduling) (haute,moyenne, basse) ou le mix des deux. RTOS est capable d'interompt les taches quand il veut, ou qu'il le veut.

4 Gestion Memoire

Mem Array of Heap figure.

Les taches partages :

- HEAP
- Flash - Var globals.

Les taches ne partages pas :

- Stack.

5 Utilisation Typique Semaphore

Une des utilisation du semaphore est la gestion des ressources. On peut aussi synchronise les taches. ** faire attention car on peut ajouter autant de token que l'on veut. ** On peut faire de la synchronisation par gestion des credits.

6 Utilisation Typique Message

On utilise la methode des messages avec 3 boites. VIDE, PAS VIDE, PLEINE. soit en FIFO, FILO, file bidirectionnel, en broadcasting.

7 Analyse des Performances

voir le code pour le benchmark.