Titre : Approche formelle pour le développement de services système dans les systèmes embarqués : du modèle à l'implantation

Durée : 2 à 3 mois

Encadrants : Karine Altisen et Christophe Rippert
contacts : {Karine.Altisen, Christophe.Rippert}@grenoble-inp.fr

Lieu du stage :
Laboratoire Verimag, 2 avenue de Vignates 38610 Gières (http://www-verimag.imag.fr/)
Equipe d'accueil : Synchrone (http://www-verimag.imag.fr/Synchron.html)

Contexte scientifique

Le développement de systèmes embarqués critiques (par exemple avions, cartes à puce) nécessite l'utilisation de méthodes de conception fiables basées sur des modèles formels permettant l'utilisation d'outils de validations automatiques des programmes.

C'est dans cette optique qu'a été conçu le langage synchrone Lustre, un langage synchrone à flot de données pour la programmation de programmes réactifs. Il est doté d'une sémantique formelle ce qui permet la validation des programmes par des techniques telles que le *model-checking* ou l'interprétation abstraite.

Il permet une programmation concurrente parfaitement déterministe grâce à un calcul statique de l'ordonnancement des tâches, et traditionnellement, les applications développées en Lustre s'exécutent directement sur la machine nue.

Cependant, l'évolution récente du domaine plaide en faveur de l'introduction d'un minimum de services systèmes pour faciliter l'implantation de fonctions supplémentaires. Par exemple, l'introduction d'une dose limitée d'ordonnancement dynamique pourra permettre la gestion de tâches urgentes (e.g. interruptions) survenant de façon imprévisibles et s'exécutant pendant une durée inconnue lors de la conception du système.

Depuis quelques temps déjà, des travaux visent à introduire de l'ordonnancement dynamique dans l'implémentation de programme Lustre, tout en garantissant sa sémantique. Ces travaux, pour beaucoup théoriques, ont conduit à des expérimentations sur des systèmes d'exploitation temps-réel généralistes.

Cadre général du sujet

Le sujet que nous proposons se positionne comme la suite de ces travaux, en s'attachant aux spécificités des systèmes embarqués. En effet, dans un système embarqué, les besoins en services système sont a priori restreints et souvent la limite entre le système d'exploitation et l'application elle-même est floue. Les noyaux de système d'exploitation généralistes sont souvent monolithiques, ce qui ne permet pas la souplesse réclamées par ces spécificités (en particulier l'extraction des services système réellement nécessaires pour une application donnée). De plus, une caractéristique particulière des systèmes embarqués est qu'ils sont en général dédiés à une seule application : un système généraliste conçu pour multiplexer les ressources matérielles entre de nombreuses applications n'est donc pas du tout adapté à ce contexte d'utilisation.

Nous proposons d'utiliser une architecture de système d'exploitation de type "exo-noyau". Cette architecture se caractérise par la séparation entre un nano-noyau contenant uniquement la couche d'abstraction du matériel et un ensemble de services système développés au niveau applicatif. Ces services système sont conçus de façon modulaire pour permettre de n'inclure que ceux absolument nécessaires à l'exécution des applications. De plus, cette architecture permet de minimiser la portion de code qui sera exécutée en mode superviseur, ce qui limite l'impact sur le système global d'une déficience d'un des services système.

Sujet

Le sujet consiste mettre en place le cadre permettant le développement de librairies dédiées de services système et à développer de tels services préalablement valides formellement. Précisément, cela implique :

- la programmation en Lustre de pilote de périphériques simples (e.g. UART, capteurs élémentaires) ainsi que de services système comme par exemple un ordonnanceur dynamique ;

- la validation formelle des programmes ;
- leur compilation et intégration dans l'exo-noyau.

La plate-forme Lego Mindstorm NXT nous a deja permis de mener un certains nombres d'expérimentations sur le sujet et servira de terrain d'implantation pour les travaux réalisés. Ceci en deux étapes :

1. implantation sur la plate-forme Lego Mindstorm NXT en utilisant le noyau monolithique existant et en y remplaçant les services système existants par les services systèmes valides ;
2. développement d'un nano-noyau et d'une librairie de services systèmes valides pour la plate-forme.

Compétences requises

Ce sujet nécessite un bon niveau de connaissance dans le domaine de la programmation système (e.g. programmation concurrente, développement de code bas-niveau, intégration dans le code existant écrit en C et assembleur ARM). Vu la difficulté technique du développement attendu, il s'adresse principalement à des étudiants ayant déjà participé à des projets de développement système.

Bibliographie :

[Caspi08] Paul Caspi, Norman Scaife, Christos Sofronis, Stavros Tripakis. "Semantics-Preserving Multitask Implementation of Synchronous Programs". *ACM Transactions on Embedded Computing Systems*, Vol. 7, No. 2. February 2008.

[Exokernel] Dawson R. Engler, M. Frans Kaashoek, James O'Toole Jr. "Exokernel : an operating system architecture for application-level resource management. In *Proceedings of the 15th ACM Symposium on Operating Systems Principles*, December 1995, pages 251-266.

[Think] Jean-Philippe Fassino, Jean-Bernard Stefani, Julia Lawall, Gilles Mulle. "Think: A Software Framework for Component-based Operating System Kernels". In *Proceedings of the 2002 USENIX Annual Technical Conference*, June 10-15, 2002.

[Lustre] Tutorial Lustre : http://www-verimag.imag.fr/~halbwach/PS/**tutorial**.ps