

Sujet de recherche

Une approche langage pour l'évolution des drivers dans Linux

Encadrant : Gilles MULLER, projet OBASCO EMN-INRIA, LINA
Co-encadrant : Yoann PADIOLEAU, projet OBASCO EMN-INRIA, LINA
courriel : Gilles.Muller@emn.fr
tél. : 02 51 85 82 04

Contexte scientifique

Il est aujourd'hui reconnu que les pilotes de périphériques (appelés également *drivers*) constituent la majeure partie d'un système d'exploitation généraliste tel que Linux ou Windows. Par ailleurs, il a été récemment démontré que les drivers sont à l'origine de 7 fois plus de bugs que le reste du système d'exploitation [1]. Si les drivers étaient des programmes stables, on pourrait espérer que la situation s'améliore *in fine*. Malheureusement, les drivers évoluent régulièrement : l'apparition de nouveaux matériels voire de nouveaux types de bus induisent des changements dans la structure du logiciel ; cette structure évolue également pour des raisons de performance et d'optimisation de la concurrence.

En ce qui concerne leur structure, les drivers ne sont généralement pas des entités monolithiques. Ils sont en fait constitués de multiples couches offrant des services communs (e.g., bus, protocoles, périphériques virtuels). Le noyau lui-même, offre des services de base tels que les fils de contrôle, la gestion mémoire, le contrôle de la concurrence. De fait, toute évolution des services communs affecte de multiples drivers ; ces évolutions doivent être répercutées vers l'ensemble des drivers affectés.

À l'heure actuelle, le processus de gestion de l'évolution est entièrement manuel et réside entièrement sur l'expertise des programmeurs. De fait, il en résulte des temps de déploiement parfois extrêmement longs (plus de 2 ans sur le remplacement de la primitive `check_region`), des bugs et des drivers orphelins que l'on sait difficilement mettre à niveau.

Objectif du stage

Nous avons récemment proposé une approche pour l'évolution des logiciels système patrimoniaux reposant sur la logique temporelle et des aspects de réécriture. Cette approche a été appliquée avec succès dans l'évolution du noyau Linux pour implémenter le système d'exécution de l'environnement Bossa [2, 3]. Dans le contexte de ce stage, il s'agit d'étendre cette approche de manière à permettre l'écriture de *patches d'évolution* permettant de capturer de manière formelle les évolutions qui doivent être réalisées sur un driver lorsque l'une des couches génériques dont il dépend évolue.

Travail à réaliser

Le travail à réaliser dans ce stage se situe dans le contexte du projet ANR Coccinelle qui vise à concevoir un langage pour la spécification de patches d'évolution et les outils associés pour automatiser le processus d'évolution dans les drivers [4, 5]. Dans un premier temps, le travail consistera à réaliser une bibliographie sur la problématique de l'évolution et de la maintenance de logiciels. Dans un deuxième temps, le travail consistera à identifier des exemples précis d'évolution

dans les drivers dans le noyau Linux 2.6 et à concevoir les *patches d'évolution* adéquats. L'étudiant pourra également participer à la conception du langage d'évolution ainsi qu'à son implémentation.

Des bonnes connaissances en système d'exploitation et dans le langage de programmation C sont requises. Des bonnes connaissances en compilation et dans le langage de programmation OCaml seraient un plus.

Références

- [1] Dawson R. Engler, David Yu Chen, Andy Chou, and Benjamin Chelf. Bugs as deviant behavior : A general approach to inferring errors in systems code. In *Proceedings of the 18th ACM Symposium on Operating System Principles*, pages 57–72, Banff, Canada, October 2001.
- [2] Rickard A. Åberg, Julia L. Lawall, Mario Südholt, Gilles Muller, and Anne-Françoise Le Meur. On the automatic evolution of an OS kernel using temporal logic and AOP. In *Proceedings of the 18th IEEE International Conference on Automated Software Engineering (ASE 2003)*, pages 196–204, Montreal, Canada, October 2003. IEEE.
- [3] Gilles Muller, Julia Lawall, Jean-Marc Menaud, and Mario Südholt. Constructing component-based extension interfaces in legacy systems code. In *ACM SIGOPS European Workshop 2004 (EW2004)*, September 2004.
- [4] Julia L. Lawall, Gilles Muller, and Richard Urunuela. Tarantula : Killing driver bugs before they hatch. In *The 4th AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS)*, pages 13–18, Chicago, IL, March 2005.
- [5] Yoann Padioleau, Julia L. Lawall, and Gilles Muller. Understanding collateral evolution in linux device drivers. Research report, INRIA, Rennes, France, November 2005.