

Auteurs : CHEYNET Julien – GOUIN Hélios – SÉNÉGAS Matthieu

CONTEXTE ET OBJECTIF

L'institut de Recherche Technologique (IRT) **Nanoelec**, fondé en 2012 et financé par le PIA (Programme d'Investissements d'Avenir), est un consortium de 22 partenaires dont le **CEA Leti**, un institut du CEA spécialisée en microélectronique et en nanotechnologies.



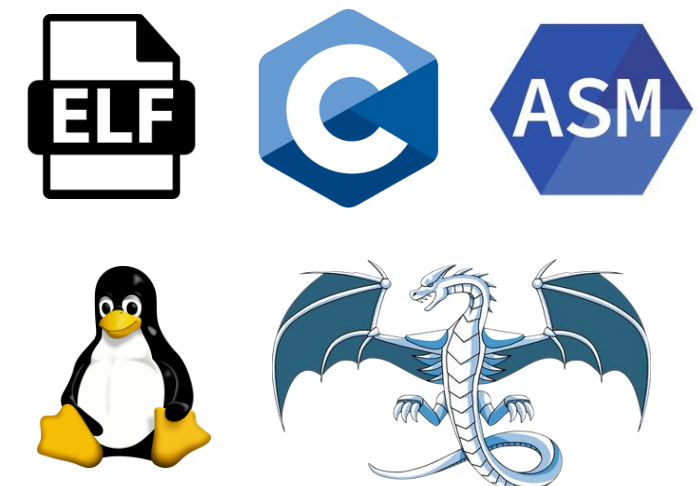
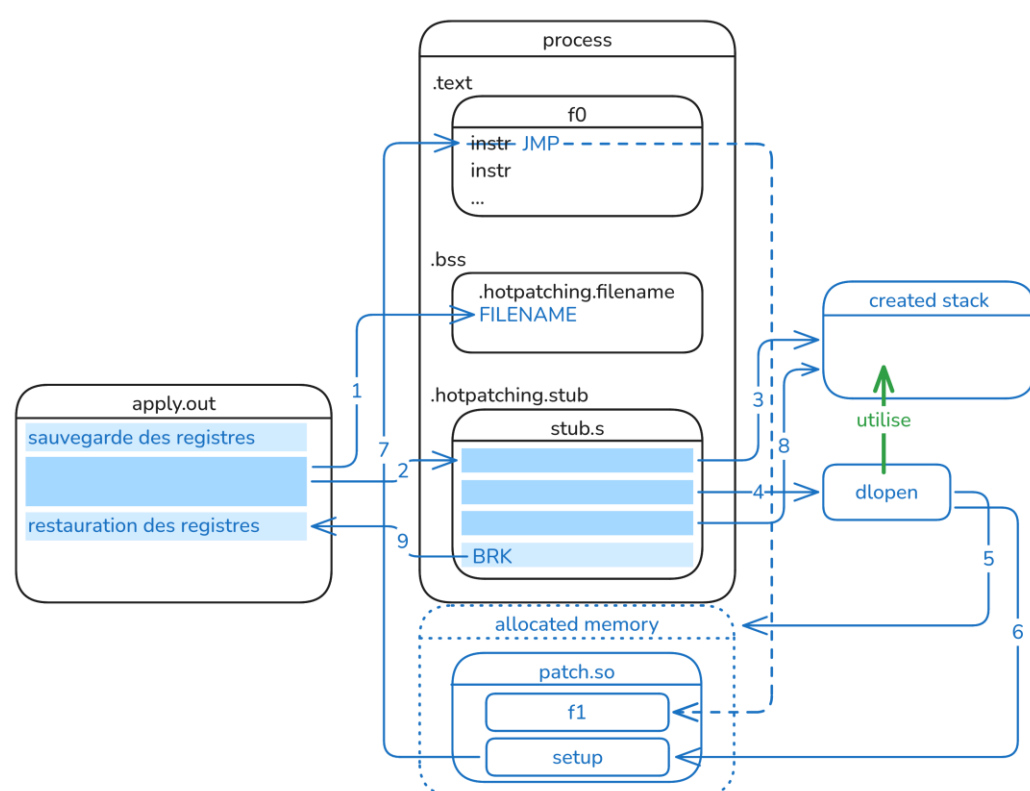
Le **hot-patching** est une pratique qui consiste à appliquer un patch **lors de l'exécution du programme**. Ce mécanisme peut être particulièrement intéressant dans le cas des systèmes industriels qui ne doivent pas être stoppés pour ne pas perdre leur état.

La solution cherche à **s'intégrer** le plus possible dans la chaîne de compilation du développeur, et à **automatiser** tout le processus de gestion du patch afin de la rendre la plus simple possible.

MÉTHODES ET DÉVELOPPEMENTS

Le projet a été réalisé en suivant une méthode agile. Après avoir effectué un état de l'art sur ce sujet, trois grands axes ont été explorés :

- Comment appliquer le patch ?
- Comment construire le patch ?
- Quand appliquer le patch ?



Le projet a été écrit en C et a été construit de façon **modulaire** : il est divisé en multiples petits **sous-programmes** et un **driver** s'occupe de les orchestrer. La solution nécessite d'instrumenter le code à différentes étapes de la compilation et de la construction du patch (pour répondre à la question de « quand patcher » ou bien pour la gestion des données statiques par exemple), pour cela un plugin **LLVM** est utilisé.

Les développements ont été validés sur un **STM32mp1**, un micro-processeur ARM 32-bit, intégrant un OS Linux embarqué.

RÉSULTATS ET CONCLUSION

La solution développée permet de patcher des logiciels complexes comme **Valkey** ou **Redis**, validée par un framework de **tests unitaires** qui vérifie toutes les fonctionnalités de la solution :

- patch de code
- patch de données (globales et statiques)
- multiples versions de patches

Il y a tout de même quelques limitations :

- patch de données allouées dynamiquement (tas)
- le linkage dynamique reste nécessaire (pas statique)
- certaines optimisations ne sont pas compatibles (LTO)
- un overhead de mémoire à chaque patch

Le hot-patching reste un sujet de recherche !



MOTS-CLÉS : hot-patching, ARM, Linux, ELF, LLVM