

**Avis de Soutenance**

Madame Darine RAMMAL

Informatique

Soutiendra publiquement ses travaux de thèse intitulés
*Sécurité de la mémoire pour la programmation réactive synchrone*
dirigés par Monsieur PIERRE RETY

Ecole doctorale : Mathématiques, Informatique, Physique Théorique et Ingénierie des Systèmes - MIPTIS
Unité de recherche : LIFO - Laboratoire d’Informatique Fondamentale d’Orléans

***Soutenance prévue le vendredi 19 janvier 2024 à 14h00***Lieu :   LIFO Batiment 3IA, Rue Léonard de Vinci, 45067 ORLEANS
Salle : Amphi Herbrand

**Composition du jury proposé**

|  |  |  |
| --- | --- | --- |
| Mme Wadoud BOUSDIRA  | Laboratoire d'Informatique Fondamentale d'Orléans (LIFO, EA 4022)  | Co-encadrante de thèse  |
| M. Julien SIGNOLES  | CEA LIST  | Rapporteur  |
| M. Fabrice MOURLIN  | Université Paris Est Créteil (UPEC)  | Rapporteur  |
| M. Frédéric  LOULERGUE  | Laboratoire d'Informatique Fondamentale d'Orléans (LIFO, EA 4022)  | Examinateur  |
| M. Abdelmalek BENZEKRI  | Institut de Recherche en Informatique de Toulouse (IRIT)  | Examinateur  |
| M. Frédéric GAVA  | Laboratoire d'Algorithmique, Complexité et Logique (LACL), Université Paris-Est Créteil (UPEC)  | Examinateur  |
| M. Frédéric DABROWSKI  | Laboratoire d'Informatique Fondamentale d'Orléans (LIFO, EA 4022)  | Invité |
| **Mots-clés :**  | Programmation réactive, Ordonnancement coopératif, Sécurité de la mémoire, Parallélisme, Rust, Système de type, |

|  |
| --- |
| **Résumé :**   |
| Les langages réactifs synchrones constituent un excellent choix pour la programmation de l'IoT en raison de leur sémantique claire pour l'interaction entre le système et l'environnement. Cependant, en ce qui concerne les systèmes critiques en termes de sécurité ou contraints en ressources, des propositions récentes telles que Fairthreads ou ReactiveML font face à un problème bien connu. La gestion manuelle de la mémoire de Fairthreads peut entraîner des erreurs, tandis que la collecte des déchets de ReactiveML assure la sécurité de la mémoire mais introduit une surcharge d'exécution. Cette thèse vise à résoudre le problème de la sécurité de la mémoire en développant un langage de programmation réactif spécifiquement conçu pour les systèmes en temps réel, intégrant des threads coopératifs et une exécution synchrone. En nous appuyant sur le solide système de types du langage de programmation Rust, nous proposons un système de types similaire à Rust pour un langage de programmation réactif de noyau appelé MSSL. MSSL offre un modèle de threads coopératifs et facilite le partage de données mutables entre les threads tout en préservant la sécurité des types et des emprunts. Pour y parvenir, nous introduisons une nouvelle abstraction appelée Trc (Thread Reference Counting), qui combine la sécurité de propriété des références de Rust avec le mécanisme de comptage de références des pointeurs intelligents de Rust. Nous présentons ensuite la sémantique et le système de types de MSSL pour démontrer ses capacités à maintenir la sécurité des types, la sécurité des emprunts et la sécurité de la concurrence. De plus, nous étendons MSSL en introduisant des extensions réactives, en intégrant le concept de signaux qui offrent des moyens de communication puissants, flexibles et fiables. Enfin, nous fournissons une implémentation en Java de l'ensemble complet de MSSL, en basant sur la sémantique et les règles de typage de son système de types. |
|   |