

1| CURRICULUM VITÆ

Khaoula BOUKIR

DOCTORANTE EN INFORMATIQUE À L'UNIVERSITÉ DE NANTES

LS2N, Équipe "*Systèmes Temps Réel*"

1 rue de la Nœe, 44321, Nantes

☎ +33 (0)7 81 83 01 98 ✉ khaoula.boukir@ls2n.fr



Date de Naissance : 27/06/1991 (29 ans)

Lieu de Naissance : El Jadida (Maroc)

Situation matrimoniale : Célibataire

Adresse personnelle : 9 rue du Fresche Blanc 44300, Nantes

1.1 Formation Académique

Octobre 2016 - Aujourd'hui

Thèse de doctorat en Informatique Temps Réel à l'Université de Nantes, Nantes - France.

- *Sujet de thèse* : mise en œuvre de politiques d'ordonnancement temps réel prouvées.
- *Directeur de thèse* : Jean-Luc BÉCHENNEC, Chargé de recherche CNRS, jean-luc.bechenne@ls2n.fr.
- *Co-encadrante* : Anne-Marie DÉPLANCHE, Maître de conférence à l'Université de Nantes, anne-marie.deplanche@ls2n.fr.
- *Laboratoire de rattachement* : Laboratoire des Sciences du Numérique de Nantes.
- *École doctorale* : ED MathSTIC¹

2015 - 2016

Master 2 recherche à l'École Centrale de Nantes, Nantes - France.

- *Titre* : Automatique, Robotique et Informatique Appliquée, *spécialité* : "Temps Réel, Conduite et Supervision", *parcours* : "Systèmes Temps Réel".
- *Classement* : **2ème** du tronc commun, **Major** en spécialité.

1. www.ed-mathstic.u-bretagne-normandie.fr

2012 - 2014

Formation d'Ingénieur d'état à l'École Nationale des Sciences Appliquées, Fès - Maroc.

- *Spécialité* : Génie des Réseaux et Télécommunications.
- *Classement* : **Major de la promotion.**

2009 - 2012

Licence des Sciences et Techniques à la Faculté des Sciences et Techniques, Fès - Maroc.

- *Spécialité* : Électronique, Télécommunication et Informatique.
- *Classement* : **Major de la promotion.**

1.2 Expériences

01/09/2019 - 31/08/2020

Poste d'Attachée Temporaire d'Enseignement et de Recherche en informatique.

- **Enseignement** : à l'école d'ingénieur Polytech'Nantes <www.polytech.univ-nantes.fr> - Département informatique.
- **Recherche** : Doctorante au Laboratoire des Sciences du Numérique de Nantes <www.ls2n.fr> - Équipe : Systèmes Temps Réel.

01/09/2017 - 31/08/2019

Contrat doctoral d'enseignement en électronique et informatique industrielle.

- **Enseignement** : à l'IUT de Nantes <www.iutnantes.univ-nantes.fr> - Département génie électronique et informatique industrielle.

01/09/2016 - 31/08/2017

Enseignante vacataire en informatique.

- **Enseignement** : à l'École Centrale de Nantes <www.ec-nantes.fr> - Département informatique.

01/10/2015 - 31/07/2016

Initiation à la recherche dans le cadre du Master 2 ARIA, au sein de l'équipe STR du LS2N, sous la direction de Mme Anne-Marie Déplanche.

- **Séminaire bibliographique** : Etude de la politique U-EDF [Nel+12] pour l'ordonnancement temps réel multiprocesseur.
- **Stage de recherche** : Mise en œuvre d'un ordonnanceur global dans l'OS temps réel Trampoline [Bec+06].

19/12/2014 - 19/04/2015

Ingénieur Réseaux freelance au sein du groupe Hala à Fès.

01/03/2014 - 30/06/2014

Stage de fin d'études, Huawei Technologies, Rabat, *thème* : « Optimisation de la boucle du Nord de Maroc Telecom avec la nouvelle génération WDM et intégration d'une interface mobile pour sa supervision ».

04/07/2013 - 04/09/2013

Stage d'application, Fedaso, Fès, *thème* : « Développement d'une attaque cybernétique de type déni de service sur un système d'information ».

09/04/2012 - 05/06/2012

Stage de fin d'études, Office Chérifien des Phosphates, El Jadida, *thème* : « Elaboration d'un programme pour l'automatisation et la supervision du groupe de motopompes principales ».

03/08/2010 - 02/09/2010

Stage d'initiation, Compagnie des Boissons Gazeuses du Nord, Fès.

15/06/2010 - 15/07/2010

Stage d'initiation, Office Nationale d'Electricité.

1.3 Compétences

Enseignement Programmation orientée objet - Algorithmique et programmation - Systèmes temps réel - Systèmes d'exploitation - Systèmes électroniques - Outils logiciels pour l'électronique - Logique classique - .

Recherche Systèmes temps réel - Ordonnancement temps réel - Systèmes d'exploitation temps réel - Application des méthodes formelles pour la vérification des systèmes d'exploitation temps réel - Model-checking - Implémentation de politiques d'ordonnancement temps réel.

Langages C/C++ - Python - Assembleur - HTML.

Logiciels CodeBlocks - Visual Studio - Atmel AVR Studio - Eclipse - DevC++ - Matlab (programmation), Uppaal - Roméo (model-checking), Simulink, ADS, Atoll, Cosis (simulateur de liaison électrique/télécom), Simso (simulateur d'ordonnancement multiprocesseur)

Langues Anglais, Français, Arabe (bilingue)

1.4 Publications

- **Conférences internationales avec comité de lecture**

- [BBD20] **Requirement specification and model-checking of a real-time scheduler implementation**

Khaoula BOUKIR, Jean-Luc BÉCHENNEC, Anne-Marie DÉPLANCHE
28th International Conference on Real-Time Networks and Systems (RTNS'20)

- [BBD18] **Formal approach for a verified implementation of Global EDF in Trampoline**

Khaoula BOUKIR, Jean-Luc BÉCHENNEC, Anne-Marie DÉPLANCHE
26th International Conference on Real-Time Networks and Systems (RTNS'18)

- [BBD17] **Reducing the gap between theory and practice : : towards a proven implementation of Global EDF in Trampoline**

(Best paper award)

Khaoula BOUKIR, Jean-Luc BÉCHENNEC, Anne-Marie DÉPLANCHE
11th Junior Researcher Workshop on Real-Time Computing (JRWRTC'17)
en conjonction avec RTNS'17.

- **Conférences nationales :**

- [BBD19] **Vérification d'une implémentation d'ordonnanceur temps réel par model-checking**

Khaoula BOUKIR, Jean-Luc BÉCHENNEC, Anne-Marie DÉPLANCHE
13th GDR SOC2 National Symposium (GDR SOC2 2019).

- **Comité de lecture** 12th Junior Researcher Workshop on Real-Time Computing (JRWRTC'18) en conjonction avec RTNS'18.

1.5 Activités para-académiques

- **2ème prix** au concours de « Ma thèse en 180 secondes » à la finale nantaise (février 2018)
- Représentante des doctorants du site de Nantes auprès de l'école doctorale MathSTIC Bretagne/Loire (janvier 2017 - aujourd'hui)
- Présidente de la cellule Revue du comité d'organisation de la 3ème édition de la compétition de robotique « IronBrain Competition of Mechatronics » de l'ENSA - Fès (2013).
- Vice-présidente du comité d'organisation des 1ère et 2ème éditions de la journée sportive de l'USMBA - Fès (2011 et 2012).

1.6 **Loisirs**

- Lecture.
- Voyages.
- Natation.

2 | TRAVAUX DE RECHERCHE

2.1 Travaux de recherche en thèse

2.1.1 Introduction

Ce chapitre présente les orientations des travaux de recherche menés au cours de ma thèse. Une présentation du contexte général et des motivations de mes travaux de recherche est fournie, suivie d'une brève introduction aux étapes suivies pour mener ces travaux.

Mots clés : Implémentation des ordonnanceurs temps réel, Système d'exploitation temps réel, Méthodes formelles pour la vérification des systèmes d'exploitation temps réel, Model-checking, Automates finis étendus.

2.1.2 Contexte général

Un système temps réel est un ensemble de programmes applicatifs qui se distingue par son aptitude à contrôler un environnement par nature dynamique, qu'on appelle procédé. La dénomination temps réel provient du fait que le système doit s'adapter et réagir à la vitesse de l'évolution du procédé contrôlé, et donc fournir des résultats exacts dans un intervalle de temps fixé. De ce fait, les systèmes temps réel sont confrontés à des contraintes temporelles dont le respect est considérablement important.

Un tel système est généralement composé d'un ensemble de programmes applicatifs. Pour la plupart, ils s'exécutent de manière récurrente, sur une plateforme informatique contenant un ensemble limité d'unités de traitement¹ partagées entre eux. L'exécution de certains de ces programmes peut être soumise à une date d'échéance.

La réalisation de tels systèmes s'appuie, très souvent, sur des systèmes d'exploitation dits systèmes d'exploitation temps réel (RTOS). L'architecture de ces systèmes d'exploitation doit être choisie de telle manière à ce qu'elle réponde aux exigences et complexité croissantes des applications temps réel, notamment en terme de rapidité de traitement. Afin de garantir les performances des systèmes temps réel, les RTOS possèdent une entité incontournable qui se charge d'organiser l'exécution des programmes et l'accès aux ressources de calcul tout en respectant les contraintes temporelles. Il s'agit de l'ordonnanceur [Man13].

Les performances d'un système temps réel dépendent également de l'architecture matérielle sur laquelle il est implémenté. Avec le progrès technologique, il y a une tendance croissante vers l'utilisation de plateformes constituées de plusieurs ressources de traitement de type

1. Unités de traitement : ce sont les ressources que les applications requièrent pour leur exécution, e.g. ressources CPU, ressources de communication, ressources logiques comme des données, etc.

multiprocesseur ou encore multicœur. Cette évolution a engendré un nombre accru d'études scientifiques en matière d'ordonnancement temps réel multiprocesseur.

Ainsi, de nombreuses politiques d'ordonnancement temps réel multiprocesseur ont été proposées, offrant jusqu'à l'optimalité et permettant, en théorie, une exploitation plus efficace des ressources processeur en plus d'une meilleure gestion en cas de surcharge. Cependant, peu d'attention a été portée à l'implémentation de ces politiques au sein d'une plateforme réelle, étant donné que leur description dans la littérature fait une abstraction complète des contraintes d'implémentation (choix des structures de données, mécanismes de gestion des interruptions, gestion des événements d'ordonnancement, overheads d'exécution, etc.). Ainsi, il existe toujours une grande distance entre la description abstraite des algorithmes d'ordonnancement et leur réalisation concrète.

2.1.3 Motivations

Afin de bénéficier des performances que les plateformes multicœurs modernes offrent, il est nécessaire que les RTOS migrent vers les politiques d'ordonnancement multiprocesseurs prometteuses. Cependant, pour adopter de telles politiques, il est primordial d'étudier en premier lieu la faisabilité de leur mise en œuvre au sein d'une plateforme réelle et d'évaluer leurs véritables performances. Dans cette optique, notre objectif est d'étendre Trampoline RTOS² [Bec+06] afin qu'il puisse supporter les politiques d'ordonnancement multiprocesseur global.

Tel que Trampoline est actuellement conçu, l'intégration de nouveaux ordonnanceurs est de type *kernel-based*. Cela veut dire que les fonctions relatives à l'ordonnancement sont immergées dans le code du noyau. Cependant, l'implémentation d'un ordonnanceur au niveau du noyau n'est pas une tâche facile et nécessite une parfaite maîtrise du code de l'OS. Aussi, il faut tenir compte des multiples contraintes d'implémentation qui sont complètement abstraites dans la littérature. De ce fait, la programmation des ordonnanceurs est sujette aux erreurs, d'autant plus que leurs codes sont traditionnellement écrits en langage C ou en Assembleur avec une utilisation intensive de macros ou de pointeurs, et sont généralement répartis sur différentes parties du noyau de l'OS. Ainsi, la présence d'erreurs dans une implémentation soulève la préoccupation permanente que l'ordonnanceur pourrait ne pas se comporter comme prévu.

Ceci amène de nombreuses questions : comment s'assurer que la mise en œuvre d'une nouvelle politique d'ordonnancement au sein d'un RTOS est correcte ? Comment vérifier que l'ordonnanceur produit toujours le comportement attendu conformément aux spécifications fournies en littérature ? Quels sont les moyens efficaces pour vérifier cette correction et comment établir une approche efficace pour cette vérification ? Afin d'adresser cette problématique, nous utilisons une approche formelle pour vérifier des exigences formulées d'un ordonnanceur global que nous avons implémenté au sein de Trampoline.

2.1.4 Présentation de l'approche

Notre objectif est de proposer une approche de vérification formelle, permettant de vérifier qu'une implémentation d'un ordonnanceur global est correcte et que son comportement est conforme aux spécifications. Pour cela nous utilisons une approche de vérification par model-checking d'une implémentation d'un ordonnanceur global au sein de Trampoline. Notre démarche est menée en trois étapes :

- **implémentation d'un ordonnanceur global** : Trampoline, dans sa version initiale, implémente une politique d'ordonnancement basique qui repose sur des priorités fixes au niveau des tâches, et sur du partitionnement en multiprocesseur. Cette étape consiste à

2. Système d'exploitation temps réel développé au sein de l'équipe STR du LS2N.

modifier le code de l'OS pour pouvoir supporter l'ordonnancement global. Une première politique implémentée est G-EDF qui nous permet de tester la faisabilité de notre approche de vérification. Les détails et choix d'implémentation de cette politique sont présentés dans [BBD17].

- **modélisation de l'ordonnanceur implémenté** : afin de vérifier la correction de l'ordonnanceur par model-checking, il faut élaborer des modèles abstrayant les différentes interactions et opérations de l'ordonnanceur au sein de l'OS. Nous nous sommes inspirés dans cette partie d'une thèse réalisée au sein de la même équipe par Toussaint TIGORI [TBR15] qui a proposé un modèle complet de Trampoline conçu en utilisant l'outil UPPAAL [Ben+96]. Ce modèle regroupe toutes les fonctions et les services de l'OS qui ont été ramenés à une combinaison d'automates finis étendus³ et de fonctions UPPAAL écrites dans une syntaxe similaire au langage C. Notre travail de modélisation consiste à traduire le code source de l'ordonnanceur et des autres composants du noyau qui contribuent à la décision d'ordonnancement en d'automates finis étendus et éventuellement temporisés.
- **vérification de la correction d'implémentation** : en matière de vérification formelle, l'objectif est, en tout premier lieu, d'élaborer des preuves de la correction fonctionnelle de l'ordonnanceur implémenté. Nous entendons par correction fonctionnelle la cohérence du comportement effectif produit par l'implémentation par rapport au comportement attendu. Une telle vérification envisage ainsi la possibilité de contrôler si toutes les "bonnes" propriétés que les études théoriques annoncent, sont préservées. De ce fait, notre approche de vérification vise à identifier des exigences sur la base du comportement attendu de l'implémentation. La mise en œuvre de la vérification de ces exigences est effectuée par le biais de modèles jouant le rôle d'observateurs. Ces observateurs permettent de mener la vérification directement lors de l'exécution du modèle, ce qui permet de conclure si l'ordonnanceur se comporte de manière cohérente à la spécification [BBD20].

Notre travail a fait l'objet de trois publications dans des conférences internationales (cf. 1.4). La soutenance de thèse est prévue pour la fin de l'année 2020.

2.2 Encadrement de recherche

Au cours de l'année universitaire 2016/2017 (1ère année de thèse), j'ai encadré un étudiant en M2 ARIA (Automatique, Robotique et Informatique Appliquée)⁴ dans le module *Initiation à la recherche* et j'ai assuré son suivi pendant son stage de master.

Le sujet que j'ai proposé pour ses travaux de recherche s'intitule "Le comportement périodique des ordonnanceurs temps réel sur des plateformes multiprocesseur identiques (pour tâches dépendantes)". L'objectif est d'appréhender le principe d'ordonnancement temps réel multiprocesseur, comprendre la classification des politiques d'ordonnancement multiprocesseur et étudier le comportement de certaines de ces politiques sur un intervalle de simulation qui peut se répéter périodiquement. Le travail de l'étudiant devait se baser sur deux travaux de recherche récents qui proposent un intervalle de simulation pour n'importe quel ordonnanceur déterministe et sans mémoire en multiprocesseur [GGCG16; GGCG13].

Dans la deuxième partie de son master, j'ai également assuré son suivi et son intégration dans son stage de master effectué au sein de de l'entreprise Capgemini. Sa mission consistait à proposer une solution permettant le suivi des projets menés en entreprise en temps réel.

3. Un automate fini étendu est une extension d'un simple automate fini avec un ensemble de variables. Dans ce cas, les transitions sont complétées avec des conditions et des actions sur ces variables.

4. à l'École Centrale de Nantes

BIBLIOGRAPHIE

- [Nel+12] Geoffrey NELISSEN et al. « U-EDF : An unfair but optimal multiprocessor scheduling algorithm for sporadic tasks ». In : *Real-Time Systems (ECRTS), 2012 24th Euromicro Conference on*. IEEE. 2012, p. 13-23.
- [Bec+06] Jean-Luc BÉCHENNEC et al. « Trampoline an open source implementation of the osek/vdx rtos specification ». In : *Emerging Technologies and Factory Automation, 2006. ETFA '06. IEEE Conference on*. IEEE. 2006, p. 62-69.
- [BBD20] Khaoula BOUKIR, Jean-Luc BÉCHENNEC et Anne-Marie DÉPLANCHE. « Requirement specification and model-checking of a real-time scheduler implementation ». In : *Proceedings of the 28th International Conference on Real-Time Networks and Systems*. 2020, p. 89-99.
- [BBD18] Khaoula BOUKIR, Jean-Luc BÉCHENNEC et Anne-Marie DÉPLANCHE. « Formal approach for a verified implementation of Global EDF in Trampoline ». In : *Proceedings of the 26th International Conference on Real-Time Networks and Systems*. ACM. 2018, p. 83-92.
- [BBD17] Khaoula BOUKIR, Jean-Luc BÉCHENNEC et Anne-Marie DÉPLANCHE. « Reducing the gap between theory and practice : towards A Proven Implementation of Global EDF in Trampoline ». In : *JRWRTC 2017* (2017), p. 9.
- [BBD19] Khaoula BOUKIR, Jean-Luc BÉCHENNEC et Anne-Marie DÉPLANCHE. « Vérification d'une implémentation d'ordonnanceur temps réel par model-checking ». In : *Proceedings of the 27th International Conference on Real-Time Networks and Systems*. ACM. 2019.
- [Man13] Florian MANY. « Combinaison des aspects temps réel et sûreté de fonctionnement pour la conception des plateformes avioniques ». Thèse de doct. Université de Toulouse, 2013.
- [TBR15] Kabland Toussaint Gautier TIGORI, Jean-Luc BÉCHENNEC et Olivier Henri ROUX. « Approche formelle pour la spécialisation de systèmes d'exploitation temps réel ». In : *Modélisation des Systèmes Réactifs (MSR 2015)*. 2015.
- [Ben+96] Johan BENGTSOON et al. « UPPAAL—a tool suite for automatic verification of real-time systems ». In : *Hybrid Systems III* (1996), p. 232-243.
- [GGCG16] Joël GOOSSENS, Emmanuel GROLLEAU et Liliana CUCU-GROSJEAN. « Periodicity of real-time schedules for dependent periodic tasks on identical multiprocessor platforms ». In : *Real-time systems* 52.6 (2016), p. 808-832.
- [GGCG13] Emmanuel GROLLEAU, Joël GOOSSENS et Liliana CUCU-GROSJEAN. « On the periodic behavior of real-time schedulers on identical multiprocessor platforms ». In : *arXiv preprint arXiv :1305.3849* (2013).