

1| CURRICULUM VITÆ

Khaoula BOUKIR

POST-DOCTORANTE AU CEA DE SACLAY

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

🏠 <https://khaoulaboukir.github.io/>



Nationalité : Française
Date de Naissance : 27/06/1991 (29 ans)
Lieu de Naissance : El Jadida (Maroc)
Situation matrimoniale : Célibataire
Adresse personnelle : 10 rue de chartres, 91400, Orsay

1.1 Situation actuelle et antérieure

Depuis Janvier 2021, je suis en poste de recherche post-doctorale au sein du CEA Paris-Saclay à temps plein dans le Laboratoire pour la Sûreté du Logiciel (*LSL*) du Département Ingénierie Logiciels et Systèmes (*DILS*). Je suis chargée de l'étude suivante : « Analyse approfondie pour la sécurité des codes avec Frama-C ».

L'année précédente, j'ai travaillé en qualité d'Attachée Temporaire d'Enseignement et de Recherche (ATER) à temps plein aux deux départements Informatique et PeiP (Parcours des écoles d'ingénieurs Polytech) de Polytech'Nantes.

1.2 Formation Académique

Octobre 2016 - Décembre 2020

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.

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.**

1.3 Expériences

20/01/2021 - Aujourd'hui

Poste de postdoctorante en informatique.

- **Sujet** : analyse approfondie pour la sécurité des codes avec Frama-C.
- **Superviseur** : Julien Signoles, Ingénieur de recherche au CEA, Julien.Signoles@cea.fr.

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 - Départements : informatique et PeiP.
- **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.

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

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 - Maroc, *sujet* : « 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 - Maroc, *sujet* : « 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 - Maroc, *Sujet* : « 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 - Maroc.

15/06/2010 - 15/07/2010

Stage d'initiation, Office Nationale d'Électricité, Fès - Maroc.

1.4 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/CSS.

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

Langues Anglais, Français, Arabe (bilingue)

1.5 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.

1.6 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 - Août 2019)
- 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).

1.7 Loisirs

- Lecture.
- Voyages.
- Natation.

2 | TRAVAUX DE RECHERCHE EN THÈSE

2.1 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 : Politiques d'ordonnancement temps réel, Système d'exploitation temps réel, Implémentation d'ordonnanceur, Model-checking.

2.2 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 complexités 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 multiprocesseur ou encore multicœur. Cette évolution a engendré un nombre accru d'études

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

scientifiques en matière d’ordonnancement temps réel multiprocesseur.

2.3 Motivations

En matière d’ordonnancement temps réel, de nombreuses politiques d’ordonnancement 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. Cette distance rend difficile le travail d’implémentation d’ordonnanceurs au sein d’une plateforme réelle. Elle soulève même des interrogations sur la faisabilité et/ou les performances obtenues des politiques implémentées.

De ce fait, afin d’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. Dans cette optique, notre objectif est de vérifier des implémentations d’ordonnanceurs globaux au sein du système d’exploitation temps réel Trampoline² [Bec+06].

Tel que Trampoline est 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. Un tel travail est potentiellement sujet aux erreurs, d’autant plus les fonctions de l’ordonnanceur sont traditionnellement écrites en langage C ou en Assembleur avec une utilisation intensive de macros ou de pointeurs, et sont généralement réparties sur différentes parties du noyau de l’OS. Ainsi, conduire une telle implémentation soulève la préoccupation permanente que l’ordonnanceur pourrait ne pas se comporter comme prévu.

Ceci amène de nombreuses questions : comment s’assurer que l’implémentation d’une nouvelle politique d’ordonnancement au sein d’un RTOS est correcte et produit toujours le comportement attendu conformément aux spécifications fournies en littérature ? Quelle démarche devons-nous adopter pour vérifier cette correction et comment l’établir ? En réalité, si nous souhaitons que les résultats théoriques en matière d’ordonnancement se concrétisent sous forme d’implémentations, ces dernières doivent être soutenues par des moyens de vérification fournissant une preuve de leur correction fonctionnelle. Afin de répondre à cette problématique, nous explorons l’utilisation d’une approche formelle permettant de vérifier des exigences formulées qui traduisent le comportement attendu des implémentations de politiques d’ordonnancement réalisées.

2.4 Présentation de l’approche

L’objectif de ma thèse 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. Ainsi, nous proposons une approche de vérification bien identifiée basée sur le model-checking. Nous cherchons à ce que cette approche soit la plus générique possible, la plus indépendante possible de la politique d’ordonnancement à vérifier, et quelle puisse être adaptée selon

2. Système d’exploitation temps réel conforme au standard OSEK/VDX développé au sein de l’équipe STR du LS2N.

le système d'exploitation cible et le nombre et la nature des composants liés à l'ordonnanceur. Cette approche est menée en deux grandes phases :

- **modélisation de l'ordonnanceur implémenté** : afin de vérifier la correction d'une implémentation d'ordonnanceur par model-checking, il faut élaborer un modèle décrivant 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 modélisés par une combinaison d'automates finis étendus³ et de fonctions UPPAAL écrites dans une syntaxe similaire au langage C. Notre travail a consisté à adapter le modèle initial de l'OS pour supporter l'ordonnancement global et y intégrer les modèles de l'ordonnanceur et des autres composants du noyau qui contribuent à la décision d'ordonnancement.
- **vérification de la correction d'implémentation** : en matière de vérification, nous identifions quatre étapes :
 1. traduire les propriétés littéraires liées à la politique implémentée en exigences décrivant le comportement attendu de l'implémentation ;
 2. formaliser les exigences ainsi établies sous forme de modèles d'observateur en mesure de suivre de manière non intrusive le comportement réel de l'ordonnanceur lorsqu'il est stimulé par des événements d'ordonnancement et permettant de statuer sur sa justesse ou non ;
 3. générer des événements d'ordonnancement permettant de stimuler le modèle de l'implémentation et le soumettre à des scénarios d'exécution permettant de vérifier les exigences établies ;
 4. conduire la vérification des exigences sur le modèle de l'implémentation en fonction des scénarios d'événements d'ordonnancement générés en examinant des propriétés CTL [CE81] en UPPAAL.

Cette approche a permis la vérification de la correction fonctionnelle du comportement de deux implémentations d'ordonnanceurs globaux au sein de Trampoline : G-EDF et EDF-US[ξ] [BBD20]. J'ai pu ainsi identifier et corriger des erreurs d'implémentation pour les deux politiques. Toutefois, son caractère modulaire et générique permet d'en envisager l'usage pour d'autres politiques et dans le cadre d'autres systèmes d'exploitation.

Mon travail de thèse a fait l'objet de trois publications dans des conférences internationales avec comité de lecture (cf. 1.5).

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.

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 BENGTTSSON et al. « UPPAAL—a tool suite for automatic verification of real-time systems ». In : *Hybrid Systems III* (1996), p. 232-243.
- [CE81] Edmund M CLARKE et E Allen EMERSON. « Design and synthesis of synchronization skeletons using branching time temporal logic ». In : *Workshop on Logic of Programs*. Springer. 1981, p. 52-71.