

NFP103 - Applications concurrentes : conception et outils de validation (ACCOV)

Présentation

Prérequis

Avoir le niveau licence informatique (L3).

Public concerné : Élèves ingénieurs, étudiants en master

Objectifs pédagogiques

De par le développement des technologies Web, des langages de programmation concurrente, des outils de programmation réseau et celui des processeurs multi-cœurs, le calcul concurrent est aujourd'hui omniprésent dans la construction de systèmes comme les systèmes d'exploitation, les systèmes distribués et les systèmes temps réel. Cependant, la conception de tels systèmes et la preuve de leur correction sont des tâches très difficiles.

Ce cours a pour objectif :

- d'acquérir une connaissance pratique des "bons" patrons de la programmation concurrente (Java)
- de comprendre les problèmes fondamentaux des systèmes concurrents
- et de s'initier à des méthodes et techniques de vérification automatique de ces systèmes (model-checking, logiques temporelles)

Compétences

conception, programmation et validation d'applications concurrents fiables

Programme

Contenu

Structuration des applications concurrentes

Contrôle de concurrence dans les systèmes transactionnels, les systèmes d'information répartis, les applications temps réel.

Les paradigmes de la concurrence et les archétypes de programmation ('design patterns').

Exclusion mutuelle, élection, producteur consommateur, lecteurs rédacteurs, client-serveur, "peer to peer", problèmes liés aux pannes, diffusion atomique ordonnée, inter-blocage, famine, équité, terminaison.

Mécanismes de bases (processus, sémaphores, moniteurs, la classe "thread" et les méthodes "synchronized" dans Java, tâches et objets protégés dans ADA95, communication synchrone et asynchrone, messages, boîtes aux lettres, invocation à distance, rendez-vous). Modularité et objets concurrents.

Spécification et vérification de propriétés de systèmes concurrents

Aperçu des méthodes de spécification : automates, automates synchronisés, réseaux de Petri, structures de Kripke, logiques temporelles.

Techniques d'analyse : analyse structurelle (réseaux de Petri), model-checking (Logique temporelle). Utilisation d'outils (open source) de simulation et de vérification : Spin, Design/CPN.

Modalités de validation

- Contrôle continu
- Projet(s)

Bibliographie

Titre	Auteur(s)
Principles of Concurrent and Distributed Programming , Addison-Wesley, 2006.	M. Ben-

Non valide depuis le 31-08-2021

Code : NFP103

Unité d'enseignement de type mixte

6 crédits

Volume horaire de référence (+/- 10%) : **50 heures**

Responsabilité nationale :

EPN05 - Informatique / 1

Contact national :

EPN05 - Informatique

2 rue Conté

75003 Paris

01 40 27 22 58

Swathi RANGANADIN

RAJASELVAM

swathi.ranganadin@lecnam.net

Programmation concurrente en Java. Éditions Pearson Education , Collection Référence, 2009

Brian
Goetz

Méthodes formelles pour les systèmes répartis et coopératifs (Traité IC2, série informatique et systèmes d' information); Ed Lavoisier 2006

S.
Haddad &
al