

Programmation concurrente

Domaine	Ingénierie et Architecture
Filière	Télécommunications
Orientation	Sécurité de l'information (TS)
Mode de formation	Plein temps

Informations générales

Nom:	:	Programmation concurrente
Identifiant:	:	PCO
Années académiques	:	2014-2015, 2015-2016, 2016-2017, 2017-2018, 2018-2019, 2019-2020
Responsable:	:	Yann Thoma
Charge de travail:	:	120 heures d'études
Périodes encadrées:	:	64 (= 48 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6
Cours						32			
Laboratoire						32			

Connaissances préalables recommandées

L'étudiant doit connaître et savoir utiliser les notions suivantes :

- langage C ou C++: constituants d'un programme, types simples et structurés (tableau, enregistrement), structures de contrôle, sous-programmes, unités d'encapsulation, pointeurs ;
- pseudo-code, algorithmes de base sur des structures de données linéaires (listes, files, ...).
- Notion de tâches/processus et ordonnancement.

Les unités d'enseignement INF1 et INF2 (informatique) et ASD1 (algorithmes et structures de données) permettent d'acquérir les connaissances de programmation nécessaires, et SYE les notions de tâches et ordonnancement.

Objectifs

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- situer le domaine de la programmation concurrente parmi les autres techniques de programmation existantes ;
- expliquer les notions de threads et de contexte d'exécution ;
- décrire le concept d'ordonnancement de threads ;
- expliquer, concevoir et utiliser des threads ;
- expliquer les concepts d'exclusion mutuelle, de communication et de synchronisation entre threads ;
- analyser et expliquer le fonctionnement d'un programme concurrent ;
- expliquer et utiliser la notion de sémaphore ;
- expliquer et utiliser la notion de moniteur.

A l'issue des travaux pratiques en laboratoire, principalement destinés à l'assimilation des connaissances

et à l'acquisition d'expérience dans le développement d'applications concurrentes, l'étudiant-e sera en outre capable de :

- implémenter un processus sous la forme de threads ;
- réaliser et utiliser des outils d'exclusion mutuelle et de synchronisation : verrous, sémaphores, moniteur ;
- simuler des outils de programmation concurrente ;
- exercer le test de programmes concurrents ;
- acquérir de l'expérience dans l'analyse de code, la détection et la correction d'erreurs de programmation concurrente.

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- Rappel de la notion de threads/ordonnancement	1
- Exclusion mutuelle : problématique, atomicité, attente active, introduction au problème de l'interblocage, famine, algorithme de Peterson.	6
- Notion de sémaphore : définition, propriétés, sémantique de base (P et V), verrous.	6
- Notion de moniteur : définition et sémantique	4
- Problèmes classiques : analyse, étude et production de solutions dans les cas du producteur-consommateur, des lecteurs-rédacteurs, traitements prioritaires.	12
- Design patterns concurrents	3

Laboratoire: 32 périodes

- Expérimentation de la nécessité de la notion d'exclusion mutuelle.	6
- Mise en oeuvre (maquettes de trains miniatures) des notions de base de synchronisation et de gestion de ressources partagées.	8
- Mise en oeuvre (maquettes de trains miniatures) de notions avancées de communication, de synchronisation et de gestion de ressources partagées.	6
- Réalisation d'une application utilisant les sémaphores comme outils de synchronisation.	6
- Réalisation d'une application utilisant les moniteurs comme outils de synchronisation.	6

Bibliographie

Aucune information

Contrôle de connaissances

Cours:

l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 2 périodes.

Laboratoire:

ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

Examen:

L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 90 minutes.

Matériel autorisé:

- Information communiquée directement par l'enseignant.

Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5

Fiche validée le 16.06.2015 par Sanchez Eduardo