

## Programmation concurrente

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Informatique et systèmes de communication
<b>Orientation</b>	Systèmes informatiques embarqués (ISCE)
<b>Mode de formation</b>	Temps partiel/En emploi

### Informations générales

Nom	: Programmation concurrente
Identifiant	: PCO
Année académique	: 2021-2022
Responsable	: Yann Thoma
Charge de travail	: 90 heures d'études
Périodes encadrées	: 64 (= 48 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6	E4	S7	S8
Cours								32				
Laboratoire								32				

### Connaissances préalables recommandées

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

- langage 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, ...).

Les unités d'enseignement PRG1 et ASD (algorithmes et structures de données) permettent d'acquérir les connaissances de programmation nécessaires.

### 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 la notion de thread ainsi que ce que la préemption implique dans un programme multi-threadé ;
- utiliser des threads pour décomposer un programme ;
- expliquer et utiliser 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 ;
- détecter des erreurs dans du code concurrent.

Un accent particulier sera mis sur les exercices, et notamment la revue des codes rendus, afin d'acquérir l'oeil permettant de détecter les erreurs qui peuvent se trouver dans du code concurrent. Les différentes sections du contenu du cours prennent donc en compte les parties théoriques et la correction des exercices.

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 :

- développer une application multi-threadée ;
- utiliser des outils d'exclusion mutuelle et de synchronisation : verrous, sémaphores, moniteur ;
- développer des mécanismes de type producteurs-consommateurs et lecteurs-rédacteurs ;

A l'issue des travaux pratiques en laboratoire, et des séances d'exercices, l'étudiant aura entre autre eu l'occasion de :

- 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

- Introduction aux threads et à l'ordonnancement	3
- Exclusion mutuelle : problématique, atomicité, attente active, introduction au problème de l'interblocage, famine, algorithme de Peterson.	4
- Notion de sémaphore : définition, propriétés, sémantique de base (P et V), verrous.	6
- Notion de moniteur : définition et sémantique	5
- Problèmes classiques : analyse, étude et production de solutions dans les cas du producteur-consommateur, des lecteurs-rédacteurs, traitements prioritaires.	12
- Design patterns concurrents	2

**Laboratoire:** 32 périodes

- Expérimentation de la nécessité de la notion d'exclusion mutuelle.	2
- Mise en oeuvre d'une application multi-threadée	4
- Mise en oeuvre (maquettes de trains miniatures) des notions de base 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
- Réalisation d'une application plus complexe exploitant la concurrence	8

## 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 et du code rendu, 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 :

- Aucun matériel, ni documentation autorisés

**Calcul de la note finale**

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