

Programmation concurrente et parallèle

riveill@unice.fr

<http://www.i3s.unice.fr/~riveill>



Pré-requis et objectifs

- Pré-requis
 - Savoir programmer et debugger des programmes séquentiels
 - Savoir programmer en Java (création de thread)
 - Savoir programmer en C (manipulation de pointeurs)
 - La création de thread sera abordée pendant ce cours
 - Savoir programmer en Python
 - La création de threads sera abordée pendant le projet
 - Vouloir apprendre des choses nouvelles
- Objectifs
 - Connaitre les concepts fondamentaux de la programmation concurrentes
 - Savoir modéliser des problèmes simples
 - Etre capable de mettre en œuvre les principaux schémas de synchronisation
 - Comprendre les problèmes liés à la performance/correction des programmes parallèles
 - Si ces 3 points sont acquis → UE validée (note >10)
 - Autres notions abordées
 - Savoir mettre en œuvre une preuve sur des problèmes simples
 - Savoir détecter rapidement qu'un programme est correct ou incorrect
 - Maîtriser les principes du model-checking, logique temporelle
 - Adapter les algorithmes en fonction de la gestion mémoire effectuée par le processeur
 - Si ces notions sont acquise → note > 16

Rythme et support

- Chaque vendredi
 - 1 heure de cours
 - 1 h 30 de TD
 - 1 h 30 de travail personnel
 - + surement un peu plus pour faire avancer le projet
- Quelques règles de fonctionnement
 - On arrive à l'heure ou on ne vient pas
 - Si on vient c'est pour travailler sur le cours et pas sur un autre cours
 - En contrepartie de ces contraintes
 - Pas d'appel / Pas de devoir surprise
 - Les évaluations seront annoncées une semaine en avance
- Support : [moodle \(bientôt\)](#) + [canal slack](#)
 - Polycopié (seul admis pendant les examens)
 - Copie des transparents données après le cours
 - Pendant le cours, on écoute et on prend des notes
 - Sujets des TD
 - Puis proposition de correction le lendemain du TD

Evaluation du cours

- Deux ou trois QCM pour 2/3 de la note globale
 - Toutes les notes comptent
 - Pas de rattrapage
 - Si absence justifiée → calcul avec les autres autres notes
 - Si absence non justifiée → 0 à l'épreuve
- Une note de projets pour 1/3 par binôme
 - Evaluation en lien avec le cours i.e. la programmation parallèle et concurrente

Projet

- **Le projet doit** vous permettre de mettre en évidence votre esprit d'analyse
 - A commencer dès cette semaine
 - A faire par deux, évaluation régulière au cours du semestre
 - **L'aide est autorisée, le plagiat/la recopie ne sont pas autorisés**
 - Chaque fichier de code doit avoir un auteur
 - **Objectif :** mettre en œuvre différentes notions apprises dans le cours et surtout, savoir **restituer ces notions** dans un rapport
 - Réalisation de plusieurs prototypes
 - En Python
 - Dans un notebook jupyter qui contient le code + les explications
 - Le notebook doit être vu comme un rapport exécutable avec un plan, des sections, une conclusion
 - **Le rapport est plus important que le code**
 - **Savoir rédiger, expliquer, se faire comprendre**
 - **Savoir prendre du recul, comparer et évaluer des solutions**
 - **C'est toujours le même notebook qui progresse**
 - lors des séances de TDs mais aussi en dehors
- **Les dates de progression et le respect des consignes sont impératives**

Des questions avant de démarrer ?