

## Programmation concurrente et parallèle

Groupe 20

Mardi, de 13h30 à 16h30 SB-M240 (cours)

Jeudi, de 13h30 à 15h30 PK-S1575 (atelier)

### Responsable(s) du cours

---

**Nom du coordonnateur :** TREMBLAY, Guy

**Nom de l'enseignant :** TREMBLAY, Guy

**Local :** PK-4435

**Téléphone :** (514) 987-3000 #8213

**Courriel :** tremblay.guy@uqam.ca

**Site Web :** <http://www.labunix.uqam.ca/~tremblay>

### Description du cours

---

Familiariser les étudiants avec les concepts de base et les différents paradigmes de la programmation concurrente et parallèle. Donner un aperçu de la mise en oeuvre de programmes concurrents et parallèles dans différents langages. - Architectures parallèles : architecture des processeurs (pipelines, coeurs multiples); multi-processeurs, multi-ordinateurs, grilles de calculs. - Types d'application concurrentes : à fils d'exécution multiples, parallèles, distribuées. - Synchronisation et communication : variables partagées, échange de messages. - Programmation par variables partagées : verrous, sémaphores, barrières, moniteurs. - Mesures de performance: temps, coût, accélération, efficacité. Stratégies de programmation : parallélisme itératif, récursif, de flux, de données, de sac de tâches; parallélisme de résultat, d'agenda, de spécialistes.

Préalables académiques :

Travaux en laboratoires avec divers langages - MPD, C, Java ou OpenMP.

### Objectifs du cours

---

- Objectif général = Initier les étudiant-e-s à la programmation concurrente et parallèle avec variables partagées.
- Objectifs spécifiques = À la fin du cours, l'étudiant-e devrait être capable...
  1. d'expliquer et d'utiliser les principales stratégies de programmation concurrente et parallèle;
  2. de lire des programmes écrits dans divers langages de programmation (Ruby, Java, Pthreads/C, OpenMP/C, TBB/C++, OpenCL/C);
  3. d'écrire des programmes dans un ou deux langages parallèles de haut niveau (Ruby, Java, OpenMP/C)

### Contenu du cours

---

- 1. Introduction : concurrence vs. parallélisme et les différentes formes d'applications concurrentes (multi-contextes, distribuée, parallèle).
- 2. Aperçu des architectures parallèles : classification de Flynn ; multi-processeurs et multi-ordinateurs ; mémoire partagée et mémoire distribuée ; architectures multi-contextes; processeurs multi-coeurs.

- 3. Introduction au langage Ruby.
- 4. Programmation concurrente impérative par variables partagées :
  - Concepts de base : processus, tâche, thread, synchronisation et communication, graphe de dépendances;
  - Patrons de programmation parallèle en PRuby: parallélisme fork-join, de boucles, de données, coordonnateur et travailleurs, de flux de données
  - Mécanismes de synchronisation : verrous, variables de condition et moniteurs, barrières;
  - Méthodologie de programmation parallèle et patrons d'algorithmes parallèles: approche PCAM; parallélisme de tâches, récursif, de données, de flux (filtres et pipelines).
  - Exemples de langages : Ruby/PRuby, C/pthread, Java/Java.util.concurrent, OpenMP/C, TBB/C++, OpenCL/C.
- 5. Mesures de performance : temps d'exécution, coût, travail, accélération et efficacité, loi d'Amdhal.

## Modalités d'évaluation

Descpriction	Date	Pondération
Examen intra	27 octobre	25 %
Examen final	15 décembre	30 %
Trois (3) travaux pratiques		45 %

L'utilisation de documentation personnelle est permise aux examens.

Une moyenne d'au moins 50 % aux examens est exigée pour réussir le cours. Idem pour les travaux pratiques.

Les travaux pratiques doivent être réalisés seul ou en équipe de deux (2) personnes.

Une pénalité de 10 % par jour de retard sera appliquée pour la remise des travaux.

La qualité du français sera prise en considération, tant dans les examens que dans les travaux pratiques (jusqu'à 10 % de pénalité).

### Politique d'absence aux examens

**L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.**

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen &ndash; par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : PK-3150 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs. Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

## Intégrité académique

**PLAGIAT Règlement no 18 sur les infractions de nature académique. (extraits)**

Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une infraction au sens de ce règlement.

La liste non limitative des infractions est définie comme suit :

- la substitution de personnes;
- l'utilisation totale ou partielle du texte d'autrui en la faisant passer pour sien ou sans indication de référence;
- la transmission d'un travail pour fins d'évaluation alors qu'il constitue essentiellement un travail qui a déjà été transmis pour fins d'évaluation académique à l'Université ou dans une autre institution d'enseignement, sauf avec l'accord préalable de l'enseignante, l'enseignant;
- l'obtention par vol, manoeuvre ou corruption de questions ou de réponses d'examen ou de tout autre document ou matériel non autorisés, ou encore d'une évaluation non méritée;
- la possession ou l'utilisation, avant ou pendant un examen, de tout document non autorisé;
- l'utilisation pendant un examen de la copie d'examen d'une autre personne;
- l'obtention de toute aide non autorisée, qu'elle soit collective ou individuelle;
- la falsification d'un document, notamment d'un document transmis par l'Université ou d'un document de l'Université transmis ou non à une tierce personne, quelles que soient les circonstances;
- la falsification de données de recherche dans un travail, notamment une thèse, un mémoire, un mémoire-crédation, un rapport de stage ou un rapport de recherche;
- Les sanctions reliées à ces infractions sont précisées à l'article 3 du Règlement no 18.

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html> et <http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

## Médiagraphie

U O <http://www.labunix.uqam.ca/~tremblay/INF5171/Materiel>

Divers éléments (notes de cours, exemples) seront mis à la disposition des étudiants sur ce site.

V C G.R. Andrews. **Foundations of Multithreaded, Parallel, and Distributed Programming**, Addison-Wesley, 2000.

V C D.R. Butenhof. **Programming with POSIX Threads**, Addison-Wesley, 1997

V C N. Carriero and D. Gelernter. **How to Write Parallel Programs - A First Course**, The MIT Press, 1990.

V C D.E. Culler and J.P. Singh. **Parallel Computer Architecture - A Hardware/Software Interface**, Morgan Kaufmann Publishers, 1999.

V C I. Foster. **Designing and Building Parallel Programs**, Addison-Wesley, 1995. <http://www.mcs.anl.gov/~itf/dbpp/text/book.html>

V C M. Gengler, S. Ubéda, and F. Desprez. **Initiation au parallélisme: Concepts, architectures et algorithmes**, Masson, 1996.

V C D. Lea. **Concurrent Programming in Java: Design Principles and Patterns (Second Edition)**, Addison-Wesley, 2000

V C T.G. Mattson, B.A. Sanders, and B.L. Massingill. **Patterns for Parallel Programming**, Addison-Wesley, 2005.

V C M.McCool, A.D. Robison, and J. Reinders. **Structured Parallel Programming: Patterns for efficient computation**, Morgan Kaufman, 2012.

V C S. Oaks, H. Wong. **Java Threads (Third Edition)**, O'Reilly, 2004.

V C M.J. Quinn. **Parallel Programming In C With MPI And OpenMP**, McGraw-Hill, 2003.

V C M.J. Sottile, T.G. Mattson, B.A. Sanders, and C.E. Rasmussen. **Introduction to Concurrency in Programming Languages**, Chapman and Hall/CRC, 2009.

V C B. Wilkinson and M. Allen. **Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers**, Prentice-Hall, 1999.