

COORDONNATEUR	TREMBLAY, Guy	tremblay.guy@uqam.ca	(514) 987-3000 8213	PK-4435
GROUPE	20 TREMBLAY, Guy	tremblay.guy@uqam.ca	(514) 987-3000 8213	PK-4435

Mardi et jeudi, de 13h30 à 15h00 (cours) – Mardi, de 15h30 à 17h30 (ateliers)

DESCRIPTION	<p>Familiariser les étudiants avec les concepts de base et les différents paradigmes de la programmation parallèle. Donner un aperçu de la mise en oeuvre d'algorithmes parallèles sur différents types d'architectures.</p> <p>Architectures parallèles: architecture des processeurs; organisation de la mémoire; réseaux d'interconnexion. Synchronisation et communication: variables partagées vs échange de messages. Mesures de complexité: temps, accélération, efficacité. Stratégies et méthodes de programmation: parallélisme de résultat, d'agenda, de spécialistes. Paradigmes non impératifs: programmation fonctionnelle, acteurs, programmation logique. Études de diverses applications: tris, fouilles, matrices, graphes, optimisation combinatoire. Travaux en laboratoire.</p> <p>Préalables: INF3172 Principes des systèmes d'exploitation</p>
-------------	---

OBJECTIFS	<ul style="list-style-type: none"> • Initier les étudiant-e-s à la programmation concurrente et parallèle. • À la fin du cours, l'étudiant-e devrait être capable: <ul style="list-style-type: none"> <input type="checkbox"/> d'expliquer et d'utiliser les principales stratégies de programmation concurrente et parallèle ; <input type="checkbox"/> de lire des programmes écrits dans divers langages de programmation (MPD, Java, C, MPI) ; <input type="checkbox"/> d'écrire des programmes dans un langage parallèle de haut niveau (MPD, Java).
-----------	---

ÉVALUATION	Description sommaire	Date	Pondération
	Examen intra	Mardi 21 octobre, 13h30 à 16h30	25%
	Examen final	Mardi 9 décembre, 13h30 à 16h30	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.

Les travaux pratiques, sauf exception avec entente préalable avec le professeur, doivent être réalisés par 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

Un étudiant absent à un examen se verra normalement attribuer la note zéro pour cet examen. Cependant, si l'étudiant était dans l'impossibilité de se présenter à l'examen pour un motif valable, certains arrangements pourront être pris avec son enseignant. Pour ce faire, l'étudiant devra présenter à son enseignant l'un des formulaires prévus à cet effet accompagné des pièces justificatives appropriées (par ex., attestation d'un médecin que l'étudiant était dans l'impossibilité de se présenter à l'examen pour des raisons de santé, lettre de la Cour en cas de participation à un jury).

Une absence pour cause de conflit d'horaires d'examen n'est pas considérée comme un motif valable d'absence, à moins d'entente préalable avec la direction du programme et l'enseignant durant la période d'annulation des inscriptions avec remboursement : tel qu'indiqué dans le guide d'inscription des étudiants, il est de la responsabilité d'un étudiant de ne s'inscrire qu'à des cours qui ne sont pas en conflit d'horaire.

Pour plus de détails sur la politique d'absence aux examens du Département d'informatique et pour obtenir les formulaires appropriés, consultez le site web suivant :

<http://www.info.uqam.ca/enseignement/politiques/absence-examen>

CONTENU	<ol style="list-style-type: none"> 1. Introduction : les différentes formes de programmation concurrente (multi-contextes, distribuée, parallèle) ; les principaux paradigmes de programmation concurrente (communication par variables partagées et par échanges de messages). 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. Programmation concurrente impérative : <ul style="list-style-type: none"> • Notions de base : processus, tâche, <i>thread</i>, synchronisation et communication ;
---------	---

- Programmation par variables partagées : verrous, sémaphores et barrières, moniteurs ;
 - Programmation par échanges de messages : communication asynchrone ; RPC et rendez-vous ;
 - Stratégies de conception d'algorithmes parallèles : parallélisme itératif, parallélisme récursif, filtres et pipelines, parallélisme de données, sac de tâches ;
 - Exemples de langages : MPD, C (Pthreads), Java (Java.util.concurrent), MPI.
4. Mesures de performance : temps d'exécution, coût, travail, accélération et efficacité, lois d'Amdhal et de Gustafson-Barsis

RÉFÉRENCES

- V O G.R. Andrews – *Foundations of Multithreaded, Parallel, and Distributed Programming* – Addison-Wesley, 2000. COOP-UQAM (Sciences).
- U O <http://www.info2.uqam.ca/~tremblay/INF5170/>
Divers éléments du cours seront mis à la disposition des étudiants sur ce site.
- V C [But97] D.R. Butenhof – *Programming with POSIX Threads* – Addison-Wesley, 1997
- V C [CG90] N. Carriero and D. Gelernter – *How to Write Parallel Programs - A First Course* – The MIT Press, 1990.
- V C [CS99] D.E. Culler and J.P. Singh – *Parallel Computer Architecture - A Hardware/Software Interface* – Morgan Kaufmann Publishers, 1999.
- V C [CT93] M. Cosnard and D. Trystram – *Algorithmes et architectures parallèles* – InterEditions, 1993. [QA76.58C68].
- V C [Fos95] I. Foster – *Designing and Building Parallel Programs* – Addison-Wesley, 1995. – <http://www-unix.mcs.anl.gov/dbpp>
- V C [GUD96] M. Gengler, S. Ubéda, and F. Desprez – *Initiation au parallélisme -- Concepts, architectures et algorithmes* – Masson, 1996. [QA76.58G45].
- V C [Lea00] D. Lea – *Concurrent Programming in Java - Design Principles and Patterns (Second Edition)* – Addison-Wesley, 2000.
- V C [MSM05] T.G. Mattson, B.A. Sanders, and B.L. Massingill – *Patterns for Parallel Programming* – Addison-Wesley, 2005.
- V C S. Oahs, H. Wong – *Java Threads (Third Edition)* – O'Reilly, 2004.
- V C [Pac97] P.S. Pacheco – *Parallel Programming with MPI* – Morgan Kaufman Publ., 1997.
- V C [Qui03] M.J. Quinn – *Parallel Programming In C With MPI And OpenMP* – McGraw-Hill, 2003.
- U C [WA99] B. Wilkinson and M. Allen – *Parallel Programming - Techniques and Applications Using Networked Workstations and Parallel Computers* – Prentice-Hall, 1999.
- U C <http://www.info2.uqam.ca/~tremblay/chercher-reference.cgi>
Une bibliographie plus détaillée est disponible à l'URL suivant (sélection <<Architecture et programmation parallèle>>).

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé