

COORDONNATEUR	TREMBLAY, Guy	tremblay.guy@uqam.ca	(514) 987-3000 8213	PK-4165
GROUPE	10 TREMBLAY, Guy	tremblay.guy@uqam.ca	(514) 987-3000 8213	PK-4165
Lundi, de 13h30 à 16h30 (cours) – Mercredi, de 9h30 à 11h30 (ateliers)				

DESCRIPTION

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

- OBJECTIFS**
- Initier les étudiant-e-s à la programmation concurrente et parallèle avec variables partagées.
 - À la fin du cours, l'étudiant-e devrait être capable:
 - d'expliquer et d'utiliser les principales stratégies de programmation concurrente et parallèle;
 - de lire des programmes écrits dans divers langages de programmation (MPD, Java, Pthreads/C, OpenMP/C);
 - d'écrire des programmes dans un ou deux langages parallèles de haut niveau (MPD, Java, OpenMP/C).

ÉVALUATION

Description sommaire	Date	Pondération
Examen intra	Mercredi 14 novembre 2012, 9 h 30 à 11 h 00	20 %
Examen final	Lundi 17 décembre 2012, 13 h 30 à 16 h 30	40 %
Deux (2) travaux pratiques		40 %

L'utilisation de documentation personnelle (papier) 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

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/reglements/politique-dabsence-aux-examens>

- CONTENU**
1. Introduction : les différentes formes de programmation concurrente (multi-contextes, distribuée, parallèle) ; les principaux paradigmes de programmation concurrente (communication par variables partagées; par échanges de messages; parallélisme de données, de contrôle, de flux).
 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 par variables partagées :
 - Notions de base : processus, tâche, *thread*, synchronisation et communication;

- Communication et synchronisation : verrous, sémaphores et barrières, variables de condition et moniteurs;
 - 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), OpenMP/C.
4. Mesures de performance : temps d'exécution, coût, travail, accélération et efficacité, lois d'Amdahl et de Gustafson-Barsis

RÉFÉRENCES

- UO <http://www.info2.uqam.ca/~tremblay/INF5171/>
Divers éléments (notes de cours, exemples) seront mis à la disposition des étudiants sur ce site.
- VC G.R. Andrews – *Foundations of Multithreaded, Parallel, and Distributed Programming* – Addison-Wesley, 2000. COOP-UQAM (Sciences).
- VC [But97] D.R. Butenhof – *Programming with POSIX Threads* – Addison-Wesley, 1997
- VC [CG90] N. Carriero and D. Gelernter – *How to Write Parallel Programs - A First Course* – The MIT Press, 1990.
- VC [CS99] D.E. Culler and J.P. Singh – *Parallel Computer Architecture - A Hardware/Software Interface* – Morgan Kaufmann Publishers, 1999.
- VC [CT93] M. Cosnard and D. Trystram – *Algorithmes et architectures parallèles* – InterEditions, 1993. [QA76.58C68].
- VC [Fos95] I. Foster – *Designing and Building Parallel Programs* – Addison-Wesley, 1995. – <http://www-unix.mcs.anl.gov/dbpp>
- VC [GUD96] M. Gengler, S. Ubéda, and F. Desprez – *Initiation au parallélisme -- Concepts, architectures et algorithmes* – Masson, 1996. [QA76.58G45].
- VC [Lea00] D. Lea – *Concurrent Programming in Java - Design Principles and Patterns (Second Edition)* – Addison-Wesley, 2000 [QA76.73538L42.2000]
- VC [MSM05] T.G. Mattson, B.A. Sanders, and B.L. Massingill – *Patterns for Parallel Programming* – Addison-Wesley, 2005 [QA76.642M38.2005]
- VC S. Oahs, H. Wong – *Java Threads (Third Edition)* – O'Reilly, 2004.
- VC [Qui03] M.J. Quinn – *Parallel Programming In C With MPI And OpenMP* – McGraw-Hill, 2003 [QA76.73C15 Q55.2004]
- UC [WA99] B. Wilkinson and M. Allen – *Parallel Programming - Techniques and Applications Using Networked Workstations and Parallel Computers* – Prentice-Hall, 1999 [QA76.642W53 1999]
- UC <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