



Département d'informatique
IFT630 – Processus concurrents et parallélisme

Plan de cours
Hiver 2014

Enseignant**Gabriel Girard**

Courriel :	gabriel.girard@usherbrooke.ca
Local :	D4-1010-3
Téléphone :	(819) 821-8000 poste 63033
Site :	http://info.usherbrooke.ca/GabrielGirard/cours/
Disponibilité :	Sur rendez-vous

Professeur responsable : Gabriel Girard

Horaire

Exposé magistral :	Lundi	14h30 à 15h20	salle D3-2034
	Jeudi	13h30 à 15h20	salle D3-2034

Description officielle de l'activité pédagogique¹

Objectifs	Se familiariser avec les concepts de la programmation concurrente. Apprendre à résoudre des problèmes en se servant de la programmation concurrente.
Contenu	Approfondissement des concepts de processus et de fil d'exécution (thread). Synchronisation centralisée ou répartie : problématique, techniques et erreurs typiques. Communication pour systèmes centralisés et pour systèmes répartis : problématique et techniques de mise en oeuvre. Architecture des systèmes de processus communicants (client/serveur, P2P, grappes, grid, ...). Coordination de processus.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 6 heures de travail personnel par semaine
Préalable	IFT 320
Particularités	Aucune

1. http://www.usherbrooke.ca/programmes/cours/IFT/ift_630.htm

1 Présentation

Cette section présente les objectifs spécifiques et le contenu détaillé de l'activité pédagogique. Cette section, non modifiable sans l'approbation d'un comité de programme du Département d'informatique, constitue la version officielle.

1.1 Mise en contexte

À l'heure où les unités de calcul se multiplient au coeur des processeurs, où les coprocesseurs, graphiques en particulier, deviennent de plus en plus puissants, il est nécessaire de pouvoir analyser un projet logiciel de façon parallèle, concurrente ou distribuée. Bien que les objectifs premiers de telles techniques soient le gain de vitesse et l'optimisation de l'utilisation des ressources matérielles, la maîtrise de ces concepts fournit au programmeur de nouveaux et précieux outils.

Ce cours demande avant tout une certaine aisance en programmation et une capacité d'analyse rigoureuse et méthodique. Des connaissances en systèmes d'exploitation, réseaux et architecture des ordinateurs seront aussi très utiles.

1.2 Objectifs spécifiques

L'objectif du cours est de permettre à l'étudiant de se familiariser avec les concepts de la programmation concurrente, les concepts utilisés sur les systèmes distribués ainsi que les notions de programmation fiable avec reprise. À la fin du cours, l'étudiant devra être capable de résoudre des problèmes en se servant de la programmation concurrente. En particulier, il devra être capable de transformer un programme séquentiel en un programme parallèle en utilisant les différents outils de synchronisation et de communication disponibles. Il devra aussi être capable de développer des programmes comportant de la reprise au niveau du système d'exploitation.

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera donc capable :

1. de maîtriser la terminologie et les concepts sous-jacents à la programmation concurrent;
2. de comprendre le fonctionnement des algorithmes de synchronisations;
3. de développer des programmes concurrents et de les synchroniser avec les divers outils de synchronisation;
4. de comprendre le fonctionnement de diverses structures de calcul sur un réseau d'ordinateurs;
5. de comprendre et appliquer les différentes techniques de parallélisation de programmes séquentiels;
6. de développer des programmes fonctionnant sur un réseau d'ordinateurs ou sur une grappe de calcul;
7. de comprendre le fonctionnement des systèmes distribués;
8. de comprendre les principes de fiabilité et de performance associés à ces systèmes;

1.3 Contenu détaillé

Thème	Contenu	Heures	Objectifs	Travaux
1	Notions de base : – Processus:concept, opérations et relations; fil d'exécution (thread): structure, terminologie, exemples; noyau : fonction, primitives, structure, implantation.	2	1	
2	Concurrence et parallélisme : – Définitions, types de concurrences, opérations pour la concurrence (fork, join, cobegin/coend, etc), modélisation de la concurrence.	2	1	
3	Synchronisation : – Définitions : synchronisation, communication, types de synchronisation; Solutions avec attente active : Dekker, Disjkstra, Peterson et autres; Sémaphores : principes, implantation et utilisation; Exemples classiques.	8	2,3	P
4	Programmation parallèle : – Principes, régions critiques conditionnelles, moniteurs, moniteurs étendus, expressions de chemins, autres mécanismes	8	2,3	P
5	Communication inter-processus : – Types de communication : mémoire commune et passages de messages; Passages de messages : identification des interlocuteurs, synchronisation, protection et implantation.	4	2,3	P
6	Systèmes d'exploitation répartis : – Systèmes de fichiers répartis, gestion de l'UCT, gestion de la mémoire, synchronisation et communication,	10	4,7	
7	Calcul parallèle: – Problématiques et approches; modèles de parallélisation: algorithmique, architectural, communication et synchronisation; outils de programmation.	3	4,5,6	P
8	Algorithhmes parallèles: – Modèles pour algorithmique parallèle : graphes orientés acycliques (DAG), PRAM, modèle réseau, Hypercube, ...; – Performance et complexité des communications	3	4,5,6	P
9	Fiabilité : – Éviter les fautes, détection des erreurs, traitement des fautes, reprise avant et arrière.	2	8	P
10	Performance : – Introduction, mesures de performances, techniques d'évaluation, charges de travail.	2	8	

Le cours doit comprendre au moins trois travaux pratiques et un projet couvrant tous les sujets marqués d'un P dans le tableau.

2 Organisation

Cette section propre à l'approche pédagogique de chaque enseignante ou enseignant présente la méthode pédagogique, le calendrier, le barème et la procédure d'évaluation ainsi que l'échéancier des travaux. Cette section doit être cohérente avec le contenu de la section précédente.

2.1 Méthode pédagogique

Une semaine comprend trois heures de présence en classe pendant laquelle la théorie est présentée. Pour la plupart des thèmes abordés, une ou plusieurs études de cas seront présentés pour accompagner la théorie. Trois travaux pratiques et un projet sont utilisées pour approfondir vos connaissances pratiques et théoriques de la plupart des sujets abordés.

2.2 Calendrier

Semaine	Date	Thèmes	Lecture
1	06/01/14	1-2	Chap. 1 (acétates), manuel (chap. 3-4)
2	13/01/14	2-3	Chap. 2 (acétates), manuel (chap. 3-4)
3	20/01/14	3	Chap. 3 (acétates), manuel (chap. 6)
4	27/01/14	3	Chap. 3 (acétates), manuel (chap. 6)
5	03/02/14	4	Chap. 4 (acétates), manuel (chap. 6)
6	10/02/14	4	Chap. 4 (acétates), manuel (chap. 6)
7	17/02/14	4-5	Chap. 5 (acétates) , manuel (chap. 3)
8	24/02/14	Mini-test	
9	10/03/14	5-6	Chap. 6 (acétates), manuel (chap. 3 -4)
10	17/03/14	6	Chap. 6 (acétates), manuel (chap. 3 -4)
11	24/03/14	7	Chap. 7 (acétates), manuel (chap. 16, 17, 18)
12	31/03/14	8	Chap. 7-8 (acétates), manuel (chap. 16, 17, 18)
13	07/04/14	8-9-10	Chap. 8-9-10 (acétates) et notes de cours
14	16/04/14	Final	
15	22/04/14	Final	

2.3 Évaluation

Devoirs (3)	20 %
Projet	25 %
Mini-test	15 %
Final	40 %

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages² l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

Le plagiat consiste à utiliser des résultats obtenus par d'autres personnes afin de les faire passer pour sien et dans le dessein de tromper l'enseignant. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 8.1.2 du *Règlement des études*³ de l'Université de Sherbrooke. L'étudiant ou l'étudiante peut s'exposer à de graves sanctions dont automatiquement un zéro (0) au devoir ou à l'examen en question. Pour plus d'information, consultez le document décrivant le plagiat sur le site web du cours.

2. http://www.usherbrooke.ca/accueil/documents/politiques/pol_2500-008/pol_evaluation/sciences.html

3. <http://www.usherbrooke.ca/programmes/etude>

Ceci n'indique pas que vous n'avez pas le droit de coopérer entre deux équipes tant que la rédaction finale des documents et la création du programme reste le fait de votre équipe. En cas de doute de plagia, l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'il considère comme étant plagié. En cas de doute, ne pas hésiter à demander conseil et assistance à l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

TP #	Thème	Réception du travail	Remise du travail
1	Sémaphores, moniteurs et messages (SR)	13/01/14	10/02/14
2	Calcul parallèle (C, C++, MPI et OpenCL)	10/02/14	10/03/14
3	Concurrence et fiabilité (Unix et C)	10/03/14	10/04/14
Projet	Systèmes parallèles	01/14	10/04/14

Directives particulières : Le projet peut être théorique ou pratique. S'il est théorique, il doit approfondir un sujet abordé dans le cours ou aborder un sujet qui n'a pas été vu dans le cadre du cours. S'il est pratique, il peut se faire sur un ou plusieurs des quatre laboratoires du département. Le sujet du projet doit être approuvé par le professeur.

3 Matériel nécessaire pour le cours

Les acétates du cours sont disponibles sur le WEB. Le manuel sur lequel est basé le cours est celui de Silbershatz [2].

4 Références

- [1] Finkel (R. A.) : *An Operating System Vade Mecum*. Prentice-Hall, 1988.
- [2] Silberschatz (A), Galvin (P.B.) et Gagne (G.) : *Operating Systems Concepts*. Wiley, 2003.
- [3] Lister (A.M.) : *Fundamentals of operating systems (third edition)*. Springer-Verlag, 1984.
- [4] Gregory ANDREWS : *Concurrent Programming : Principles and Practice*. Benjamin-Cummings, 1991.
- [5] Gregory ANDREWS : *Foundations of Multithreaded, Parallel, and Distributed Programming*. Addison-Wesley, 2000.
- [6] Gregory ANDREWS et D. A. OLSSON : *The SR Programming Language : Concurrency in Practice*. Benjamin-Cummings, 1992.
- [7] Tanenbaum (A.S.) : *Modern Operating Systems*. Prentice-Hall, 2001.
- [8] J. BACON et T. HARRIS : *Operating Systems : Concurrent and Distributed Software Design*. Addison-Wesley, 2003.
- [9] K. CRISTIAN : *The Unix operating system*. John Wiley & Sons, 1983.
- [10] H.M. DEITEL, P.J. DEITEL et D.R. CHOFFNES : *Operating systems*. Prentice-Hall, 2004.
- [11] Nutt (G.) : *Operating Systems*. Addison-Wesley, 2003.
- [12] Talluri (M.), Hill (M.D.) et Khalidi (Y. A.) : *A New Page Table for 64-bit Address Spaces*. Proceedings of the ?? operating systems symposium, SIGOPS', 1995.
- [13] Flynn (M.J.), Gray (J.N.), Jones (A.K.), Lagally (K.), Opderbeck (H.), Popek (G.J.), Randell (B.), Saltzer (J.H.) et Wiehle (H.R.) : *Operating system : an advance course*. Springer-verlag, 1978.
- [14] Stallings (W.) : *Operating Systems*. Prentice-hall, 2001.