



6.00 crédits	30.0 h + 30.0 h	Q1
--------------	-----------------	----

Enseignants	Legay Axel ;
Langue d'enseignement	Anglais > Facilités pour suivre le cours en français
Lieu du cours	Louvain-la-Neuve
Thèmes abordés	<p>Pendant longtemps, les CPU à usage général se sont concentrés sur la prise en charge efficace d'un seul thread d'exécution. Des améliorations dans la fabrication des puces ont permis d'empaqueter plus de transistors sur la même surface d'une plaquette de silicium, et de faire fonctionner des processeurs résultants à des fréquences plus élevées. Les applications monothread s'exécuteraient simplement plus rapidement avec chaque nouvelle génération de processeur. Cette ère est maintenant terminée. L'industrie a connu plusieurs limitations connues sous le nom de mur de puissance, de mur de mémoire et de mur d'ILP. Les fabricants de puces ne sont plus en mesure d'augmenter verticalement les processeurs prenant en charge un seul thread d'exécution, ils ont commencé à regrouper plusieurs unités d'exécution plus simples, ou coeurs. L'exploitation de la puissance de plusieurs coeurs nécessite l'exploitation du parallélisme dans les applications utilisant plusieurs threads. L'écriture de code concurrent nécessite l'identification et la gestion de la concurrence, et l'introduction de la synchronisation nécessaire pour l'exactitude. L'écriture de code concurrent évolutif et performant nécessite de comprendre ce compromis entre la synchronisation et le parallélisme et de maîtriser des implémentations efficaces de structures de données partagées et d'algorithmes pour une exécution simultanée. Enfin, les processeurs multicoeurs emploient une disposition de mémoire complexe, et l'hypothèse de temps d'accès mémoire uniformes n'est plus valide. Comprendre l'impact des accès mémoire non uniformes (NUMA) est donc important pour écrire du code efficace pour les processeurs multicoeurs.</p> <p>Ce cours fournira aux étudiants les outils et les connaissances nécessaires pour écrire un code efficace et évolutif pour les processeurs multicoeurs modernes. Il détaillera les mécanismes disponibles pour la synchronisation, depuis l'implémentation de constructions de langage telles que verrous, moniteurs ou variables de condition, aux utilisations directes des primitives de synchronisation fournies par le CPU (par exemple compare-and-swap) pour construire des structures de données efficaces et évolutives . Il mettra l'accent sur les aspects de performance de la programmation multicoeur: l'impact des primitives de synchronisation, l'impact de l'accès à la mémoire non uniforme et l'impact des hiérarchies de mémoire à plusieurs niveaux. Il offrira enfin une ouverture vers l'avenir de la programmation multicoeur avec une introduction à la mémoire transactionnelle et au support de l'exécution spéculative dans les CPU modernes (par exemple Intel Haswell), et discutera du modèle d'exécution du code concurrent s'exécutant sur les GPU.</p>
Acquis d'apprentissage	<p>A la fin de cette unité d'enseignement, l'étudiant est capable de :</p> <p>Eu égard au référentiel AA du programme « Master ingénieur civil en informatique », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • INFO2.1-5 • INFO4.1-4 • INFO5.1-6 • INFO6.1, INFO6.4 <p>Eu égard au référentiel AA du programme « Master [120] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • INFO2.1-4 • INFO4.1-4 • INFO5.1-3 • INFO6.2-4 <p>1</p> <p>Eu égard au référentiel AA du programme « Master [60] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • 1SINF1.M3 • 1SINF2.1-4 • 1SINF3.1-4 • 1SINF4.1-3 • 1SINF5.2-4 <p>A l'issue de ce cours, les étudiants auront appris à développer un produit logiciel de grande taille dans des conditions semi-professionnelles. Les étudiants ayant suivi ce cours avec fruit seront capables de :</p> <ul style="list-style-type: none"> • Décrire les différences entre plusieurs modèles de développement majeurs (par exemple, cascade, itératif et agile);

	<ul style="list-style-type: none"> • Distinguer entre les phases de développement de logiciels (spécifications, architecture, conception, implémentation, validation, documentation); • Réaliser, d'une manière rigoureuse et systématique, les artefacts produits dans ces différentes phases du cycle de vie du logiciel; • Appliquer une méthodologie de développement logiciel actuellement pratiquée dans l'industrie; • Travailler efficacement dans une équipe pour développer un système logiciel de taille moyenne à grande; • Gérer la coordination et la communication entre les différents membres de l'équipe; • Interagir avec un client pour identifier ses besoins, pour clarifier les spécifications imprécises, et de prendre en compte la demande des modifications tout au long du processus de développement; • Décrire les exigences fonctionnelles d'un système logiciel en utilisant, par exemple, les cas d'utilisation ou utilisateurs des histoires; • Estimer le temps et les ressources nécessaires pour compléter un tel projet de développement de logiciels, planifier les tâches à exécuter et les livrables à produire, et assurer le respect de cette planification; • Utilisez un outil de gestion de projet pour affecter le suivi les tâches de développement logiciel; • Mettre en pratique différentes méthodes et techniques pour assurer la qualité du logiciel produit; • Comprendre les problèmes inhérents au développement de grands systèmes logiciels impliquant différentes parties prenantes et qui se composent de plusieurs composants.
Modes d'évaluation des acquis des étudiants	<p>L'évaluation du cours sera basée sur:</p> <ul style="list-style-type: none"> • La participation individuelle aux travaux de groupe et aux réunions de groupe hebdomadaires avec les tuteurs de cours; • La réalisation de trois prototypes intermédiaires avec rapports techniques correspondants; • Le rapport final, le système livré et sa documentation, ainsi que la présentation et la démonstration du produit final devant le client. <p>Attention: Étant donné que ce cours est basé sur la participation à un projet d'équipe tout au long de l'année, les coûts de projets seront automatiquement conservés en seconde session.</p>
Méthodes d'enseignement	<p>Développement (analyse, conception, implémentation, validation, documentation, intégration et déploiement) d'un grand logiciel pour un client, en équipe de 6 à 8 étudiants supervisés par un chef de projet. Des réunions hebdomadaires auront lieu avec le chef de projet, et différents prototypes et rapports doivent être produits en cours du projet.</p>
Contenu	<p>Ce projet de génie logiciel consiste en le développement (analyse, conception, implémentation, validation, documentation, intégration et déploiement) d'une application logicielle réaliste et non-triviale, si possible proposée par et avec la participation d'un client réel, dans des conditions de travail semi-professionnelles.</p> <p>Le sujet de l'application à construire est proposé par un partenaire industriel ou une organisation à but non lucratif qui participe à l'organisation de ce cours.</p> <p>Des équipes de 6 à 8 étudiants (nécessaires pour réaliser un grand projet), collaboreront, supervisés par un chef de projet.</p> <p>Des réunions hebdomadaires auront lieu avec le chef de projet (un assistant ou tuteur) pour présenter le progrès et les difficultés rencontrées, pour évaluer les alternatives, et discuter de la distribution et de la planification du travail au sein de l'équipe.</p> <p>L'application à développer sera très probablement une application web, mais le choix du langage de programmation, de l'environnement, le cadre d'application et des outils de développement dépendront des exigences du client du projet.</p>
Ressources en ligne	<p>http://moodleucl.uclouvain.be/course/view.php?id=7599</p>
Bibliographie	<p>French</p> <p>Des lectures supplémentaires seront suggérées dans le plan de cours qui décrit les produits livrables et l'organisation du projet. Les supports de cours pertinents, des slides et des informations pratiques seront accessibles sur Moodle, qui sera également le principal moyen de communication entre l'enseignant et les étudiants.</p> <p>English Additional reading material will be suggested in the course plan which describes the deliverables and organisation of the project. All relevant course material, slides and practical information will be available on Moodle, which will also be the main means of communication between the teacher and the students.</p>
Autres infos	<p>Pré-requis:</p> <ul style="list-style-type: none"> • Avoir une bonne connaissance et expérience avec les concepts de la programmation orientée objet, les algorithmes et les structures de données. • Avoir participé à l'élaboration d'un système logiciel de petite à moyenne échelle.
Faculté ou entité en charge:	<p>INFO</p>

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] : ingénieur civil en informatique	INFO2M	6		
Master [120] en sciences informatiques	SINF2M	6		
Master [60] en sciences informatiques	SINF2M1	6		