

**Département d'informatique****IFT 611 / IFT 729 – Conception de systèmes temps réel****Plan d'activité pédagogique**

Hiver 2021

Enseignant

Patrice Roy

Courriel : patrice.roy@usherbrooke.ca

Local : L1-15741

Téléphone : +1 819 821-8000 x

Disponibilités : Avant le cours, lundi matin, ou par courriel

Responsable(s) : Direction du département

Site web du cours : <https://moodle.usherbrooke.ca>

Horaire

Exposé magistral : Lundi 9h30 à 12h20 salle D3-2040

Description officielle de l'activité pédagogique¹

Cibles de formation :	Connaître et repérer les problèmes inhérents au développement de systèmes temps réel ; connaître et appliquer le traitement du temps au niveau des systèmes informatiques ; spécifier, concevoir, programmer et vérifier des systèmes temps réel.
Contenu :	Types de systèmes temps réel. Représentation du temps, contraintes de temps, horloge, synchronisation d'horloges. Formalismes utilisés dans la spécification de systèmes temps réel : machines à états, statecharts, réseaux de Petri, Graftet. Approche axiomatique de spécification de contraintes temporelles. Architecture des systèmes temps réel. Acquisition et traitement de l'information en temps réel. Modèles utilisés dans la conception de systèmes temps réel : modèles basés sur les événements, modèles basés sur les graphes, modèles des tâches, modèles des processus, modèles du contrôle. Programmation d'applications.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 6 heures de travail personnel par semaine
Préalable	IFT 320
Particularités	Aucune

¹<https://www.usherbrooke.ca/admission/fiches-cours/ift611>

1 Présentation

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

1.1 Mise en contexte

Les systèmes temps réel (STR) constituent un sujet à la fois important et méconnu. Assimilés, souvent à tort, à l'idée de systèmes très rapides, ou à celle d'interface personne/machine répondant instantanément aux demandes d'un usager, les STR sont perçus par le grand public à travers une lorgnette floue, inexacte.

Pourtant, un STR n'est pas un jouet : on attend d'un STR à ce qu'il respecte certaines contraintes de « performance », la « performance » étant exprimée en temps d'exécution de tâches précises, et ce même dans le pire cas. Un STR n'est pas tant un système rapide globalement qu'un système en tout temps assez rapide (ou jamais trop lent, ce qui revient au même).

Parfois, la contrainte temps réel (TR) à respecter sera simple à exprimer : une pulsation au pire chaque nième quantum de temps ; une écriture sur disque d'une quantité donnée d'information à chaque seconde dans le pire cas ; au minimum six lectures de touches au clavier par seconde ; un rafraîchissement complet d'écran au moins chaque soixantième de seconde ; etc. Évidemment, exprimer et garantir sont deux verbes de nature bien différente.

La conception des STR constitue un sujet plus complexe qu'il n'y paraît sur les plateformes contemporaines, du fait que la majorité d'entre elles sont multiprogrammées. Il est en général impossible d'offrir des garanties assez fortes pour qu'un système puisse être qualifié de STR sur une plateforme comme Microsoft Windows ou Linux, par exemple, ce qui ne signifie pas que les plateformes ne permettent pas l'écriture de code « performant » dans l'ensemble. Ceci met en relief la distinction entre TR souple, qui rejoint l'idée que se fait la majorité des gens des STR (quelque chose d'instantané en apparence), et TR strict, qui est d'un autre ordre : le non-respect de contraintes TR souples entraîne un désagrément, mais le non-respect de contraintes TR strictes peut entraîner des bris matériels ou des blessures.

Le mode des STR est un monde de praticiens, même chez les théoriciens du domaine. Règle générale, un STR correspond à un besoin réel, défini de manière précise et mesurable. Un STR, surtout s'il est strict, a pour particularité que si ses contraintes ne sont pas rencontrées, le fruit de son travail n'est pas valide. Les résultats de l'exécution d'un STR dépendent autant de la validité de ses calculs que du respect de ses contraintes de temps. C'est d'ailleurs la définition canonique, en quelque sorte, des STR ; nous explorerons aussi d'autres définitions, d'autres visions du sens à accorder au terme STR.

Enfin, les STR ont ceci de particulier qu'ils s'inscrivent systématiquement dans une démarche architecturale : une modification à un endroit, par exemple pour fins d'optimisation, peut entraîner un non-respect des contraintes ailleurs dans le même système.

Optiques retenues

La tradition des STR est associée de près à celle des systèmes à haute performance, même s'il n'existe pas d'adéquation stricte entre les deux mondes. Certaines tâches TR (souples) peuvent être réalisées sur des plateformes grand public (Microsoft Windows, une distribution conventionnelle de Linux, peu importe), mais d'autres tâches TR (strictes) requièrent un véritable SETR. Nous partagerons notre temps entre ces deux types de plateformes.

Il est presque impossible de discuter de systèmes contemporains complexes sans adresser la question de la multiprogrammation. Nous devons donc lui accorder une place de choix.

Le monde de l'informatique contemporaine accorde une place importante aux systèmes pris en charge, qui sont intrinsèquement indéterministes sur le plan performance (ces systèmes utilisent des moteurs de collecte automatique d'ordures). Nous prendrons soin d'examiner comment une plateforme comme Java peut prétendre se décliner sous une forme TR, et à quel prix.

1.2 Cibles de formation spécifiques

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

1. De distinguer un STR d'un système qui ne l'est pas
2. D'expliquer les caractéristiques inhérentes à un STR
3. De définir la spécification de contraintes TR
4. De mesurer le respect de contraintes TR
5. De développer un module d'un STR
6. D'utiliser des primitives d'un système d'exploitation lors de la programmation de STR

Plusieurs objectifs plus contextuels seront atteints en chemin : penser un programme dans un contexte de ressources limitées (les systèmes embarqués, par exemple, qui sont sujets à être soumis à des contraintes TR) ; modifier un moteur de collecte d'ordures (typique des STR sur plateforme prise en charge) ; appliquer des techniques d'optimisation locales et systémiques ; développer des systèmes dans une optique de résilience et de tolérance aux pannes ; etc.

1.3 Contenu détaillé

Thème	Contenu	Nbr. d'heures	Objectifs
1	Première approche : Exploration des divers sens de : temps, temps réel ; historique, problèmes et définitions ; clarification de ce qu'est (ou non) un STR ; STR et déterminisme ; STR et sécurité	3	1 et 2
2	Complexité et mesure : Brève exploration : des conteneurs standards, des itérateurs, des algorithmes standards et des garanties de complexité ; complexité vs temps ; complexité en taille ou en espace ; conteneurs et itérateurs maison et adaptateurs ; outils de mesure portables et outils de mesure non portables ; approches RAI ; risques et bénéfices ; quoi mesurer	3	2, 3, 4, 5
3	Contraintes : Contraintes strictes ou souples ; contraintes locales ou systémiques ; conséquences d'un non-respect ; respect préventif (externe) ; respect disciplinaire (interne) ; scrutation ; interruptions forcées ; respect et matériel ; respect local vs contraintes d'ensemble	3	1, 2, 3, 4
4	Multiprogrammation : Fils d'exécution vs processus (bref) ; implémentations Win32 et POSIX ainsi que std : thread et ses variantes ; fonctions résumables et enjeux TR ; synchronisation et prévisibilité ; mémoire partagée ; algorithmes sans verrous et progression ; vision générale ou disciplinaire ; systèmes multi cœurs ; déterminisme et multiprogrammation	6	1, 2, 3, 4, 5, 6
5	Enjeux propres aux SETR : Environnement de développement ; outils de synchronisation ; mécanismes et métaphores de communication ; processus et états ; ordonnanceur et commutation ; tests d'ordonnancement ; gestionnaire d'interruptions ; variantes tel que autres SETR, exécutifs TR et solutions maison	6	1, 2, 3, 4, 5, 6
6	Entrées/ sorties : Contraintes TR et E/S ; traitement de signal TR ; approches non bloquantes ; approches multiprogrammées ; impact systémique des E/S ; E/S très rapides ; vitesse locale ou équilibre systémique ; équilibre et espace	3	1, 2, 3, 4, 5, 6
7	Idiomes et schémas de conception : Relation et implémentation d'idiomes et de schémas de conception aux STR tel que singleton, observateur, pImpl, proxy, ordonnanceur, etc.	3	1 et 5
8	Gestion de la mémoire : Allouer ou construire ; déclinaisons de new et de delete ; surcharge de new et delete ; allocation positionnelle ; enjeux d'alignement ; allocation assistée ; arénas ; allocateurs	6	3 et 4
9	Exécutifs : Contexte (disciplinaire) ; avantages et risques ; concevoir un exécutif ; droits et obligations ; démonstration détaillée	3	1, 3, 4
10	Systèmes pris en charge : Déterminisme ; « performance » ; Java TR ; collecte automatique d'ordures ; état de la recherche	6	1, 2, 3, 4, 5, 6
11	Systèmes embarqués : Systèmes embarqués (si le temps le permet)	3	

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

Pour favoriser l'intégration des nombreux concepts au menu, nous suivrons essentiellement le modèle suivant :

- Exposés magistraux en classe, où les étudiant(e)s sont fortement encouragés à contribuer par leurs questions et commentaires
- Travaux pratiques et exercices à teneur formative, qui permettront aux étudiant(e)s de mesurer concrètement leur compréhension de la matière explorée, et qui pourront être corrigés par le professeur ou autocorrigés à l'aide d'une grille de vérification, selon le cas
- Travaux pratiques à teneur sommative, évalués par le professeur en fonction des mêmes critères que ceux appliqués dans le cadre des activités formatives. Ces travaux seront soumis à un échéancier de livrables (voir le document Consignes quant au projet)
- Des questions de réflexion (et parfois à saveur technique) sur une base quasi hebdomadaire
- Un contrôle théorique récapitulatif, en toute fin de parcours, vérifiant formellement l'atteinte des objectifs

Certains aspects plus généraux du développement informatique seront abordés au passage : par exemple, nous chercherons non seulement à rédiger des modules réalisant le travail demandé, mais aussi du code à la fois performant et portable (la portabilité du code n'est pas tant un critère spécifique aux STR qu'un atout pour votre carrière).

Les questions et contrôles présumeront que chaque membre d'une équipe a contribué activement à la réalisation de chacun des travaux pratiques, et a bien compris les implications philosophiques et techniques de ces travaux.

J'utiliserai le SETR qu'est QNX Neutrino, un système essentiellement conforme à la norme POSIX, pour illustrer certains concepts associés à ces systèmes d'exploitation. Bien que je ne l'aie pas fait installer lors des dernières sessions (les étudiantes et les étudiants ne semblaient pas en avoir envie), sachez que je peux le faire si vous le souhaitez.

2.2 Calendrier

Semaine	Date	Thème	Notes
1	2021-01-11	1 et 2	
2	2021-01-18	2	
3	2021-01-25	3	
4	2021-02-01	4	
5	2021-02-08	5	
6	2021-02-15	5 et 6	
7	2021-02-22	Examen périodique	Je donne normalement un cours (avec l'accord des étudiantes et des étudiants) la semaine des examens, car je n'ai pas recours à un examen intra (j'utilise des minitests quasi-hebdomadaires)
8	2021-03-01	Relâche	
9	2021-03-08	7	
10	2021-03-15	8	
11	2021-03-22	8	
12	2021-03-29	8	
13	2021-04-05	9	
14	2021-04-12	10 et 11	
15	2021-04-19	Examen final	

Je donne normalement un cours (avec l'accord des étudiantes et des étudiants) la semaine des examens, car je n'ai pas recours à un examen intra (j'utilise des minitests quasi-hebdomadaires). Aussi, le calendrier tel que rempli est incorrect, mais je ne suis pas (pour le moment) capable d'indiquer des rubriques correctes pour le cours dû à des limites techniques (l'auteur de l'outil a été

informé par courriel)

2.3 Évaluation

Minitests quasi-hebdo	40 %
Travail de session	30 %
Examen final	30 %

Je passe plusieurs minitests et je conserve les huit meilleurs par étudiant(e). Chaque minitest vaut 5 points sur la note finale, pour un total de 40%.

Le travail de session est séparé en trois livrables valant respectivement 5%, 10% et 15% de la session, pour un total de 30%.

2.3.1 Qualité de la langue et de la présentation

Conformément à l'article 17 du règlement facultaire d'évaluation des apprentissages² l'enseignante ou 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.

2.3.2 Plagiat

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'enseignante ou l'enseignant. Vous trouverez en annexe un document d'information relatif à l'intégrité intellectuelle qui fait état de l'article 9.4.1 du Règlement des études³. Lors de la correction de tout travail individuel ou de groupe une attention spéciale sera portée au plagiat. Si une preuve de plagiat est attestée, elle sera traitée en conformité, entre autres, avec l'article 9.4.1 du Règlement des études de l'Université de Sherbrooke. L'étudiante ou l'étudiant peut s'exposer à de graves sanctions qui peuvent être soit l'attribution de la note E ou de la note zéro (0) pour un travail, un examen ou une activité évaluée, soit de reprendre un travail, un examen ou une activité pédagogique. Tout travail suspecté de plagiat sera transmis au Secrétaire de la Faculté des sciences. 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 restent le fait de votre équipe. En cas de doute de plagiat, l'enseignante ou l'enseignant peut demander à l'équipe d'expliquer les notions ou le fonctionnement du code qu'elle ou qu'il considère comme étant plagié. En cas d'incertitude, ne pas hésiter à demander conseil et assistance à l'enseignante ou l'enseignant afin d'éviter toute situation délicate par la suite.

2.4 Échéancier des travaux

Les dates de remise des travaux seront indiquées sur les énoncés.

2.4.1 Directives particulières

Il n'y a pas de dates préétablies pour les minitests, mais j'en passe habituellement une dizaine et il n'y en aura pas lors des deux premières semaines de la session, pas plus que lors de la période des examens finaux

Le détail du travail de session sera donné dans un document annexe.

2.5 Utilisation d'appareils électroniques et du courriel

Selon le règlement complémentaire des études, section 4.2.3⁴, l'utilisation d'ordinateurs, de cellulaires ou de tablettes pendant une prestation est interdite à condition que leur usage soit explicitement permise dans le plan de cours.

Dans ce cours, l'usage de téléphones cellulaires, de tablettes ou d'ordinateurs est autorisées. Cette permission peut être retirée en tout temps si leur usage entraîne des abus.

Tel qu'indiqué dans le règlement universitaire des études, section 4.2.3⁵, toute utilisation d'appareils de captation de la voix ou de l'image exige la permission de la personne enseignante.

²https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/Etudiants_actuels/Informations_academiques_et_reglements/2017-10-27_Reglement_facultaire_-_evaluation_des_apprentissages.pdf

³<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

⁴https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Intranet/Informations_academiques/Sciences_Reglement_complementaire_2017-05-09.pdf

⁵<https://www.usherbrooke.ca/registraire/droits-et-responsabilites/reglement-des-etudes/>

Note : L'utilisation du courriel est recommandée pour poser vos questions.

3 Matériel nécessaire pour l'activité pédagogique

Avoir un compilateur C++ à jour est une bonne idée si vous souhaitez mettre en pratique les exemples proposés en classe. Cela dit, pour le travail de session, vous aurez de la latitude en ce qui a trait aux choix technologiques

4 Références

L'intégrité intellectuelle passe, notamment, par la reconnaissance des sources utilisées. À l'Université de Sherbrooke, on y veille!

Extrait du Règlement des études (Règlement 2575-009)

9.4.1 DÉLITS RELATIFS AUX ÉTUDES

Un délit relatif aux études désigne tout acte trompeur ou toute tentative de commettre un tel acte, quant au rendement scolaire ou une exigence relative à une activité pédagogique, à un programme ou à un parcours libre.

Sont notamment considérés comme un délit relatif aux études les faits suivants :

- a) commettre un plagiat, soit faire passer ou tenter de faire passer pour sien, dans une production évaluée, le travail d'une autre personne ou des passages ou des idées tirés de l'œuvre d'autrui (ce qui inclut notamment le fait de ne pas indiquer la source d'une production, d'un passage ou d'une idée tirée de l'œuvre d'autrui);
 - b) commettre un autoplagiat, soit soumettre, sans autorisation préalable, une même production, en tout ou en partie, à plus d'une activité pédagogique ou dans une même activité pédagogique (notamment en cas de reprise);
 - c) usurper l'identité d'une autre personne ou procéder à une substitution de personne lors d'une production évaluée ou de toute autre prestation obligatoire;
 - d) fournir ou obtenir toute aide non autorisée, qu'elle soit collective ou individuelle, pour une production faisant l'objet d'une évaluation;
 - e) obtenir par vol ou toute autre manœuvre frauduleuse, posséder ou utiliser du matériel de toute forme (incluant le numérique) non autorisé avant ou pendant une production faisant l'objet d'une évaluation;
 - f) copier, contrefaire ou falsifier un document pour l'évaluation d'une activité pédagogique;
- [...]

Par plagiat, on entend notamment :

- Copier intégralement une phrase ou un passage d'un livre, d'un article de journal ou de revue, d'une page Web ou de tout autre document en omettant d'en mentionner la source ou de le mettre entre guillemets;
- reproduire des présentations, des dessins, des photographies, des graphiques, des données... sans en préciser la provenance et, dans certains cas, sans en avoir obtenu la permission de reproduire;
- utiliser, en tout ou en partie, du matériel sonore, graphique ou visuel, des pages Internet, du code de programme informatique ou des éléments de logiciel, des données ou résultats d'expérimentation ou toute autre information en provenance d'autrui en le faisant passer pour sien ou sans en citer les sources;
- résumer ou paraphraser l'idée d'un auteur sans en indiquer la source;
- traduire en partie ou en totalité un texte en omettant d'en mentionner la source ou de le mettre entre guillemets ;
- utiliser le travail d'un autre et le présenter comme sien (et ce, même si cette personne a donné son accord);
- acheter un travail sur le Web ou ailleurs et le faire passer pour sien;
- utiliser sans autorisation le même travail pour deux activités différentes (autoplagiat).

Autrement dit : mentionnez vos sources
