

Université de
Sherbrooke

Département d'informatique
IFT 209 – Programmation système
Plan d'activité pédagogique
Hiver 2024

Enseignant

Mikaël Fortin

Courriel : Mikael.Fortin@USherbrooke.ca

Local : D4-2004

Téléphone : +1 819 821-8000 x66106

Disponibilités : Sur rendez-vous

Responsable(s) : Gabriel Girard

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

Horaire

Exposé magistral :	Jeudi	8h30 à 10h20	salle D3-2041
Exercices/laboratoires :	Mercredi	10h30 à 12h20	salle D4-1023/D4-1017

Description officielle de l'activité pédagogique¹

Cibles de formation :	Comprendre l'architecture d'un ordinateur, les systèmes de numération, les types élémentaires de données, les structures de contrôle, les entrées-sorties ; savoir effectuer une programmation-système.
Contenu :	Introduction à l'architecture des ordinateurs. Système de numération. Modes d'adressage. Format des instructions machine. Représentation des données. Technique de mise au point de programmes. Arithmétique entière. Arithmétique à virgule flottante. Manipulation de bits. Sous-programmes. Application à une architecture contemporaine. Entrées-sorties. Traitement des interruptions.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure d'exercices par semaine 5 heures de travail personnel par semaine
Préalable	IFT 159
Particularités	Aucune

¹<https://www.usherbrooke.ca/admission/fiches-cours/ift209>

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

À propos des langages d'assemblage . . .

Lorsque les premiers ordinateurs sont apparus, vers la fin des années 1940, la programmation s'effectuait en langage machine, car à cette époque les outils de programmation étaient rudimentaires ou inexistants. La programmation constituait une tâche très fastidieuse puisque toutes les instructions et les données devaient être codées manuellement à l'aide d'une suite de chiffres 0 et 1. La moindre erreur pouvait entraîner plusieurs jours de travail additionnels.

Dans le but de réduire le travail des programmeurs, les informaticiens de l'époque inventèrent des langages de programmation. Les premiers furent les langages d'assemblage qui existent encore de nos jours. Ils facilitent l'écriture de programmes en utilisant une notation symbolique dans laquelle les instructions machine, les données, ainsi que les adresses de la mémoire principale sont représentées par des noms symboliques. Des modifications peuvent désormais être apportées aux programmes sans devoir les récrire complètement, les suites de chiffres 0 et 1 étant remplacées par des noms symboliques. Peu à peu les langages d'assemblage ont fait place à des langages de plus en plus évolués qui ont contribué à une amélioration sensible de la production de programmes.

Même si aujourd'hui les programmeurs écrivent rarement des programmes en langage d'assemblage, il n'en demeure pas moins que ces derniers sont encore utiles. D'une part, ils servent principalement dans la construction de compilateurs ainsi que dans la mise en oeuvre de systèmes d'exploitation et de systèmes embarqués. D'autre part, ils sont un outil précieux dans l'introduction et l'expérimentation des notions de base en architecture des ordinateurs. Enfin, ils permettent de mieux saisir les concepts fondamentaux des langages de programmation.

À propos des entrées/sorties et des interruptions . . .

De tout temps, la lenteur des entrées/sorties par rapport à la vitesse du processeur a posé le problème de l'interaction entre le processeur et les périphériques d'entrées/sorties. Une solution adéquate à ce problème nécessite la compréhension des mécanismes matériels et logiciels de synchronisation, en particulier le mécanisme d'interruption qui permet la gestion d'événements internes aux activités du processeur ou d'événements externes au processeur. La compréhension du mécanisme d'une interruption est indispensable à l'étude des principes sous-jacents aux systèmes d'exploitation.

À propos de la place de cette activité pédagogique dans votre programme . . .

L'activité pédagogique intitulée *Programmation système* est une activité pédagogique de base. Elle est préalable aux activités pédagogiques obligatoires « IFT 320 - Systèmes d'exploitation » et « IFT 585 - Télématique » des programmes d'informatique et d'informatique de gestion. Elle est également préalable, dans le programme d'informatique à l'activité pédagogique « Architecture et organisation des ordinateurs ». Sans en être un préalable, elle est utile à l'activité pédagogique à option « IFT 580 - Compilation et interprétation des langages » du programme d'informatique.

1.2 Cibles de formation spécifiques

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

1. de maîtriser l'arithmétique dans plusieurs systèmes de numération ;
2. de connaître et d'expliquer des notions de base en architecture des ordinateurs ;
3. de connaître et d'expliquer des types élémentaires de données comme les entiers, les nombres en virgule flottante, les tableaux, les chaînes de bits et les chaînes de caractères ;
4. de connaître et d'expliquer des structures de contrôle comme la séquence, la sélection, l'itération et les sous-programmes ;
5. d'utiliser des méthodes élémentaires dans le développement de programmes ;
6. de comprendre les concepts élémentaires des langages procéduraux à travers ceux des langages d'assemblage ;
7. d'écrire de petits programmes (100 à 350 instructions) dans un langage d'assemblage ;
8. de connaître et d'expliquer les opérations de lecture et d'écriture sur un périphérique d'entrées/sorties ;
9. de comprendre le mécanisme d'interruption et les différentes étapes de sa gestion.

1.3 Contenu détaillé

Thème	Contenu	Nbr. d'heures	Objectifs	Travaux	Lectures
1	Introduction : présentation de la programmation en langage d'assemblage.	2			
2	Systèmes de numération : écriture des nombres dans un système de numération ; conversion de nombres d'un système de numération en un autre.	2	1	✓	[2] chapitre 1
3	Architecture des ordinateurs : architecture von Neumann ; mémoire principale, processeur, registres ; jeux d'instructions ; aspects particuliers à l'organisation d'un ordinateur .	4	2		[2] chapitre 2
4	Accès aux données : données, adresses, modes d'adressage ; étapes de la vie d'un programme.	4	2	✓	[2] chapitre 3
5	Programmation en langage d'assemblage : survol à partir d'un petit programme ; différence entre une instruction et une pseudo-instruction ; programmation de haut niveau des entrées/sorties.	4	3, 4, 5 et 7	✓	[2] chapitre 4
6	Les nombres entiers : représentations des entiers signés et non signés ; report et débordement ; instructions arithmétiques.	4	3, 6 et 7	✓	[2] chapitre 5
7	Tableaux : tableaux à une dimension et à deux dimensions.	2	3, 6 et 7	✓	[2] chapitre 6
8	Structures de contrôle : condition, branchements, séquence, sélection, itération ; appel et retour de sous programmes ; notion d'environnement ; récursivité.	4	4, 6 et 7	✓	[2] chapitre 7
9	Chaînes de bits : opérations logiques, décalages.	2	3, 6 et 7	✓	[2] chapitre 8
10	Chaînes de caractères : codes de représentation de caractères ; opérations sur les caractères et les chaînes de caractères.	2	3, 6 et 7	✓	[2] chapitre 9
11	Sous-programmes : appel et retour, passage de paramètres, sauvegarde/récupération, récursivité.	3	4, 6 et 7	✓	[2] chapitre 10
12	Les nombres en virgule flottante : représentations des nombres en virgule flottante ; erreur d'arrondi, erreur de troncation, dépassement de capacité ; norme IEEE 754 ; instructions arithmétiques.	4	3, 6 et 7	✓	[2] chapitre 11
13	La programmation des entrées/sorties : interrogation de bits d'états, interruption, niveaux de priorités des interruptions ; illustration à l'aide de dispositifs simples.	8	8 et 9	✓	[2] chapitres 12-13

1. Le cours doit comprendre au moins six travaux pratiques couvrant tous les sujets marqués «✓» dans le tableau.
2. Les lectures indiquées ne sont là qu'à titre indicatif. L'enseignant est libre de choisir un autre document de référence.

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

L'activité pédagogique *Programmation système* est avant tout orientée vers la compréhension du fonctionnement des ordinateurs à travers l'écriture de programmes en utilisant des langages de programmation proches de la machine, en particulier les langages d'assemblage. C'est pour cette raison qu'elle exige de la part des étudiantes et des étudiants une participation active. Basée sur le manuel de cours [2] qui couvre la totalité de la matière, à l'exception du dernier thème, une semaine typique se présente sous la forme suivante :

1. Lecture par les étudiantes et les étudiants des parties du livre du cours portant sur la matière à assimiler.
2. Présentation sommaire par la personne enseignante des points les plus difficiles de la matière (environ une heure).
3. Résolution de problèmes en classe par la personne enseignante en guise de démonstration de la matière (environ une heure).

Toutes les notions théoriques relatives aux différents thèmes sont illustrées à l'aide d'exemples construits à partir de l'architecture ARM version 8 et de programmes en langage d'assemblage ARM version 8. De plus, l'architecture Nintendo NES est utilisée pour illustrer la programmation des entrées/sorties et des interruptions.

2.2 Calendrier

Semaine	Date	Thème
1	2024-01-08	1 et 2
2	2024-01-15	2 et 3
3	2024-01-22	3 et 4
4	2024-01-29	4 et 5
5	2024-02-05	5
6	2024-02-12	7
7	2024-02-19	6 et 7
8	2024-02-26	Examen périodique
9	2024-03-04	Relâche
10	2024-03-11	6
11	2024-03-18	8
12	2024-03-25	9 et 10
13	2024-04-01	11 et 12
14	2024-04-08	12 et 13
15	2024-04-15	Examen final
16	2024-04-22	Examen final

2.3 Évaluation

Devoirs (6)	25 %
Laboratoires (6)	15 %
Examen intra	25 %
Examen final	35 %

Les laboratoires sont conçus pour être réalisés durant la séance de laboratoire. L'évaluation concerne principalement le fonctionnement correct du code remis, vérifié au moyen de tests.

Les devoirs sont des travaux plus longs (1 à 3 semaines) où l'évaluation comprend le fonctionnement du code soumis avec des tests, mais également, de la qualité de la production (forme du code, choix d'implémentation).

Les exigences précises sont présentées sur les énoncés des travaux et laboratoires.

Des remises en retard sont admissibles, moyennant une pénalité.

Les examens visent à évaluer la compréhension des notions théoriques, ainsi que la capacité à comprendre et produire du code rapidement.

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

Devoirs	Sujet	Réception	Remise	Points
Devoir 1	Systèmes de numération	À définir	À définir	4
Devoir 2	Programmation ARM	À définir	À définir	4
Devoir 3	Tableaux et matrices	À définir	À définir	4
Devoir 4	Appels Système	À définir	À définir	4
Devoir 5	Récursivité	À définir	À définir	5
Devoir 6	Entrées/Sorties	À définir	À définir	4

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.

Note : Je réponds aux questions posées par courriel à l'extérieur des périodes de cours.

²https://www.usherbrooke.ca/sciences/fileadmin/sites/sciences/documents/Etudiants_actuels/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/Etudiants_actuels/Etudiants_actuels/Informations_academiques_et_reglements/Sciences_Reglement_complementaire.pdf

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

3 Matériel nécessaire pour l'activité pédagogique

Le manuel de cours [2] est fortement recommandé. L'architecture ARM version 8 est entièrement décrite dans la référence [1] . Tous les programmes et les diapositives présentés dans le cours sont disponibles sur le site moodle du cours.

Les programmes du livre [3] écrits dans la version 8 de l'architecture ARM sont également disponibles sur le site moodle du cours et sur la machine virtuelle du cours dans les laboratoires.

4 Références

- [1] ARM LIMITED : *ARMv8 Instruction Set Overview*. ARM Limited, 2011. PDF disponible sur le lecteur réseau public.
- [2] M. BLONDIN : *IFT209 – Programmation système : notes de cours*. UdeS, 2019. Recueil de notes disponible sur le lecteur réseau public.
- [3] R. ST-DENIS : *L'architecture du processeur SPARC et sa programmation en langage d'assemblage*. GGC, 2015. Livre disponible sur le lecteur réseau public.

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
