

PLAN DE COURS

Programme : 420.B0 Techniques de l'informatique

Département : Informatique

Titre du cours : ***PROGRAMMATION SYSTÈME***

Code du cours : 420-3A6-ST

Pondération : 2 - 4 - 2

Unités : 2,66

Session : Session 3

Préalables absolus : 420-2A6-ST : Programmation orienté objet
243-8B3-ST : Électronique : ordinateurs et réseaux

Préalable relatif : Aucun

Corequis : Aucun

Enseignant.e : OLIVIER FORTIN

Bureau : H-112

Coordonnées : olivier.fortin@cstjean.qc.ca et par Teams

Disponibilités : Durant les heures de cours/disponibilités prévues à l'horaire, par courriel et par Teams.

Enseignant.e : DANY GAGNON

Bureau : H-207

Coordonnées : dany.gagnon@cstjean.qc.ca et par Teams

Disponibilités : Durant les heures de cours/disponibilités prévues à l'horaire, par courriel et par Teams.

1. DESCRIPTION DU COURS

Le cours « Programmation système » (420-3A6-ST) permet aux étudiantes et étudiants du programme Techniques de l'informatique d'acquérir les connaissances et de développer les habiletés et les attitudes relatives à la programmation d'applications informatiques complexes et nécessitant une compréhension de l'ordinateur et des systèmes d'exploitation.

Les cours suivants permettent de compléter le développement de la compétence « Effectuer l'installation et la gestion d'ordinateurs » (00Q1)

- Session 2 - 243-8B3-ST - Électronique : ordinateurs et réseaux
- Session 3 - 420-3A6-ST - Programmation système

Les cours suivants permettent de compléter le développement de la compétence « Effectuer des opérations de prévention en matière de sécurité de l'information » (00Q8)

- Session 3 - 420-3A3-ST - Introduction à la réseautique
- Session 3 - 420-3A6-ST - Programmation système

Le cours Programmation système (420-3A6-ST) est également préalable aux cours *Méthodologies de la programmation* (420-4A6-ST) et *Programmation réseaux* (420-4A4-ST) offerts à la session 4.

2. OBJECTIFS DU COURS

Au terme de ce cours, l'étudiant.e aura développé les savoir-faire et les savoir-être ci-dessous et atteint les objectifs d'apprentissage suivants :

SAVOIRS	SAVOIR-FAIRE	SAVOIR-ÊTRE
– Identifier les composantes d'un système d'exploitation (p. ex. noyau, gestionnaire de mémoire, gestionnaire de processus, système de fichiers, appels systèmes, mécanismes inter-processus, etc.) – Utiliser un langage de programmation approprié pour la programmation système (p. ex. le langage C) – Utiliser un ou plusieurs langages de script (p. ex. Python, Bash, CMD, etc) – Utiliser des appels systèmes – Mettre à profit l'environnement d'exécution d'un processus : variables d'environnement, paramètres de programme, mesure de temps précises, etc. – Comprendre et utiliser un standard de paramètres structurés (p.ex. getopt) et les opérateurs de redirection – Identifier et expérimenter les failles de sécurité applicatives (p. ex. injection de code et de commandes, débordements et autres vecteurs d'attaques)	– Reconnaître les similitudes et les différences entre les langages – Reconnaître les similitudes et les différences entre les systèmes d'exploitation – Programmer le traitement de paramètres structurés – Mettre à profit une architecture multi-cœur – Choisir les bons mécanismes de synchronisation – Optimiser du code au niveau de la mémoire ou de l'exécution – Sécuriser une application – Créer et planifier des tâches systèmes (p. ex. cron) – Automatiser des tâches systèmes	– Autonomie – Capacité d'adaptation – Rigueur intellectuelle – Souci du détail – Souci de la précision – Ouverture face à la variété technologique

3. LIENS DU COURS AVEC LES ENGAGEMENTS ÉDUCATIFS¹

Les engagements éducatifs sont des objectifs de développement liés aux visées de la formation collégiale contribuant à la formation des étudiantes et étudiants. Ils font partie intégrante des fondements du plan stratégique du Cégep. Ces engagements doivent se retrouver sur le parcours de chacune et chacun au Cégep, tant en classe que dans les activités hors classe.

Les quatre engagements éducatifs sont :

- Cultiver la curiosité et le goût d'apprendre;
- Se découvrir, découvrir le monde, s'ouvrir à l'autre;
- Éduquer à l'éco responsabilité;
- Favoriser un contexte d'apprentissage interdisciplinaire.

Ce cours ne s'inscrit pas dans le développement des engagements éducatifs.

¹ CÉGEP SAINT-JEAN-SUR-RICHELIEU. *Plan stratégique 2019-2024*, Cégep Saint-Jean-sur-Richelieu, 18 juin 2019, page 20

4. PLANIFICATION DU COURS

OBJECTIFS D'APPRENTISSAGE	CONTENU	SEMAINE
Identifier les composantes d'un système d'exploitation et préparer l'environnement de travail.	– Présentation générale de la programmation système – Histoire Unix/GNU/Linux et composantes d'un système d'exploitation – Installation de Linux dans une machine virtuelle	1
Utiliser les outils de base de la programmation système.	– Terminal, shell et outils de compilation C – Syntaxe de base du langage C : types, pointeurs, fonctions et structures de contrôle – Makefile, gcc, gdb et gef	2
Effectuer des tâches de gestion du système d'exploitation.	– Commandes Linux, utilisateurs et groupes – Permissions, fichiers spéciaux, liens symboliques et dotfiles – Variables d'environnement, SSH, find et xargs	3
Automatiser des tâches et composer des commandes shell.	– Scripts Bash : shebang, variables, conditions, boucles, fonctions et trap – Flux standard, redirections et pipelines – Git, branches, merge, rebase, résolution de conflits, grep, sed et awk	4
Structurer des programmes C robustes.	– argc, argv, codes de sortie et variables d'environnement – Gestion d'erreurs : errno, perror() et strerror() – Structures C, typedef, tableaux de structures et allocation dynamique	5
Manipuler les fichiers et comprendre leur représentation par le système.	– Appels systèmes fichiers : open(), read(), write() et close() – Permissions, stat(), lstat(), fstat(), opendir() et readdir() – Systèmes de fichiers, inodes, liens, /proc, /sys et mmap()	6
Comprendre et contrôler l'exécution des processus.	– Programme vs processus, états, PID et relation parent/enfant – fork(), wait(), exit(), getpid() et famille exec() – Signaux, sigaction(), sigprocmask() et observation avec strace	7
Valider les apprentissages de mi-session et introduire la communication entre processus.	– Examen intra – Pipes anonymes et communication parent/enfant – Introduction aux mécanismes IPC Unix	8
Utiliser les mécanismes de communication inter-processus.	– FIFOs avec mkfifo() – Mémoire partagée POSIX et System V – Files de messages, sémaphores nommés, ipcs et strace	9
Comprendre la mémoire d'un processus et diagnostiquer les erreurs courantes.	– Layout de l'espace d'adressage : text, data, bss, heap et stack – RAM, stockage, pagination et mémoire virtuelle – malloc(), free(), fuites mémoire et valgrind	10

OBJECTIFS D'APPRENTISSAGE	CONTENU	SEMAINE
Analyser les exécutables et la liaison des programmes.	– Format ELF, en-têtes, sections, symboles et readelf – Liaison statique et dynamique, bibliothèques partagées et relocation – PLT, GOT, LD_PRELOAD et hooks	11
Mettre à profit une architecture multi-cœur.	– Threads POSIX avec pthread – Mutex, sémaphores, variables de condition et pthread_rwlock_t – Thread pool, atomiques C et problèmes de concurrence	12
Créer des services et appliquer des mécanismes de sécurité Linux.	– systemd, systemctl, journalctl et fichiers .service – SUID, SGID, capabilities avec setcap() et getcap() – seccomp, namespaces et isolation de processus	13
Analyser les vulnérabilités applicatives de bas niveau.	– Débogage avancé : gdb, watchpoints, catch syscall, core dumps, strace et ltrace – Assembleur x86_64, convention System V et pile – Buffer overflow, stack canaries et divulgation d'information	14-15
Exploiter et mitiger des vulnérabilités dans un contexte contrôlé.	– Python et pwntools pour l'exploitation – Format strings, ASLR, PIE, NX, ROP, ret2libc et ret2plt – Heap exploitation : use-after-free, double-free, tcache poisoning et CTF final	14-15

L'ensemble des cours de la session utilisera chaque semaine les stratégies d'apprentissage suivantes :

- Exercices dirigés (de type pas-à-pas)
- Exercices formatifs
- Appropriation d'exemples
- Lectures et visionnements
- Exposés magistraux
- Démonstrations

5. ÉVALUATION SOMMATIVE DES APPRENTISSAGES

5.1. Sommaire des évaluations sommatives

Activités d'évaluation*	Durée	Semaine	Pondération
Exercices		Chaque semaine	50 %
Examen mi-session	2 période(s)	Semaine 8	10 %
Épreuve synthèse de cours	3 périodes	Semaine 16	40 %
Total :			100 %

* Les dates des évaluations sont inscrit sur moodle pour chaque groupe

5.2. Épreuve synthèse de cours

Le tableau ci-dessous précise la forme, le contexte de réalisation, la pondération et les critères de performance de l'épreuve synthèse de cours. Cette évaluation doit permettre de valider le niveau d'atteinte des principaux objectifs d'apprentissage du cours par l'étudiant.e.

DESCRIPTION	CRITÈRES DE PERFORMANCE OU D'ÉVALUATION**	PONDÉRATION ***
L'épreuve synthèse de cours est individuelle. Elle sera faite sur ordinateur. L'élève sera questionné sur les différentes notions vues en classe et devra les appliquer à la résolution de problèmes. L'épreuve synthèse de cours aura lieu pendant la semaine d'examens. Sa durée sera de trois (3) périodes.	Analyse juste de la mise en situation et du problème. Capacité à reconnaître un problème de programmation système ou de sécurité et proposer un correctif Capacité à utiliser les appels systèmes, outils Linux et mécanismes de sécurité appropriés	40%
Total :		40 %

6. RÈGLES PARTICULIÈRES

6.1. Applications² de la *Politique institutionnelle d'évaluation des apprentissages* par le département d'informatique

– Note et règles de passage (article 5.3.2)

La note de passage est de 60 %. Cependant la note de l'épreuve synthèse de cours doit être au moins 50 %, dans le cas contraire, la note finale du cours sera au maximum 55 %.

Un travail de programmation dans lequel se trouvent une ou plusieurs erreurs de syntaxe ne peut engendrer une note supérieure à 40 %.

² CÉGEP SAINT-JEAN-SUR-RICHELIEU. *Applications de la Politique institutionnelle d'évaluation des apprentissages par les départements, la Direction de la formation continue et du Collège militaire royal de Saint-Jean, Saint-Jean-sur-Richelieu, Cégep Saint-Jean-sur-Richelieu, Ajouter année de référence (mis à jour annuellement) (Pour les étudiant.es - Voir portail du Cégep, communauté : Étudiants, onglet : Politiques)*

– **Qualité de la langue (article 5.3.3)**

Une pénalité de 0,5 % est retranchée pour chaque faute d'orthographe ou de syntaxe à compter de la première faute jusqu'à concurrence d'une pénalité maximale de 10 % de la valeur de l'épreuve.

– **Présentation des travaux (article 5.3.4)**

Une pénalité pouvant aller jusqu'à 10 % pourrait être appliquée aux travaux et exercices pour les manquements à la présentation matérielle ou méthodologique des travaux.

– **Remise des travaux (article 5.3.5)**

Les modalités suivantes viennent s'ajouter à l'article 5.3.5 de la PIEA :

b) Tout retard dans la remise des travaux entraîne une pénalité de 10 % lorsque le travail est remis à l'intérieur du délai de 24 heures.

c) Aucun travail ne sera accepté après ce délai et la note zéro (0) sera attribuée.

d) De plus, aucun travail en retard ne sera accepté pour les situations suivantes :

- Si des rétroactions individuelles ou de groupe ont été envoyées
- Si la solution ou une partie de la solution a été distribuée ou présentée en classe
- Si le travail s'est fait et remis en classe

L'étudiante ou l'étudiant doit faire des copies de sécurité afin d'éviter tout retard dû à des problèmes de matériel.

– **Modalité de reprise (article 5.3.7)**

Aucun droit de reprise n'est accordé sauf pour des situations exceptionnelles.

– **Plagiat et fraude (article 5.3.8)**

L'article 5.3.8 *Plagiat et fraude* de la PIEA s'applique.

De plus, pour les cas de plagiat ou de fraude, l'enseignante ou l'enseignant se réserve le droit de faire une évaluation orale lorsqu'il y a doute.

– **Présence au cours (article 5.4.1)**

L'article 5.4.1. Enseignement et apprentissage de la PIEA s'applique tel quel.

6.2. Autres politiques, règlement et mesures d'aide

Les étudiantes et les étudiants sont invités à consulter les documents suivants sur le site Web du Cégep :

- a) [Règlement 17](#)
- b) [Politique visant à contrer toute forme de violence ou de harcèlement](#)
- c) [Politique visant à prévenir et à combattre les violences à caractère sexuel](#)
- d) [Étudiants en situation de handicap : référence aux mesures d'aide](#)