



IFT 320 Systèmes d'exploitation

Plan de cours
Automne 2016

Enseignant : Mikaël Fortin
Courriel : mikael.fortin@usherbrooke.ca
Local : D4-2004
Téléphone : (819) 821-8000 poste 66106
Site : Lecteur réseau public, teamboard (www.teamboardapp.com)
Disponibilité : à déterminer

Auxiliaire : _____
Courriel : _____

Horaire : Lundi 10 h 30 à 12 h 20 D3-2038
Mercredi 8 h 30 à 12 h 20 D3-2038

Description officielle de l'activité pédagogique ¹

Objectifs	Connaître et comprendre les principes généraux, aussi bien de bas que de haut niveau, des systèmes d'exploitation. Comprendre les relations existant entre le système d'exploitation et la machine et entre le système d'exploitation et l'utilisateur. Connaître les outils de base pour la programmation parallèle.
Contenu	Retour sur les entrées-sorties et approfondissement. Gestion de l'espace secondaire. Systèmes de fichiers. Gestion des processus et des fils d'exécution (<i>threads</i>). Gestion mémoire physique et logique. Protection mémoire. Mémoire virtuelle. Concepts d'interblocage et notions de base en synchronisation. Éléments de protection et de sécurité. Études de cas.
Crédits	3
Organisation	3 heures d'exposé magistral par semaine 1 heure d'exercices par semaine 5 heures de travail personnel semaine
Préalables	IFT 159 et IFT 209

¹ <http://www.usherbrooke.ca/fiches-cours/ift320>

1 Présentation

1.1 Mise en contexte

Un élément fondamental commun à tous les appareils électroniques modernes est la présence d'un système d'exploitation, aussi simple soit-il. Il s'occupe de tâches fondamentales, comme l'initialisation du matériel et sa gestion, et fournit aux programmeurs une manière standard et transparente d'y accéder. Les systèmes modernes, fruits de 40 ans de recherche, sont des projets logiciels complexes et vastes. Cependant, la compréhension des bases de leur fonctionnement est nécessaire pour mieux appréhender de nombreux autres sujets, comme l'optimisation ou la sécurité. Ce cours est préalable à l'activité IFT 630.

1.2 Objectifs spécifiques

À la fin du cours, l'étudiante ou l'étudiant sera capable de:

1. Choisir un type de système d'exploitation selon les circonstances ;
2. Comprendre et utiliser les différentes fonctionnalités d'un système d'exploitation ;
3. Comprendre et analyser l'implantation d'un système de fichiers ;
4. Comprendre et analyser les tenants et aboutissants de la gestion des mémoires primaire et secondaire ;
5. Comprendre les mécanismes de protection liés au système d'exploitation ;
6. Comprendre et analyser les tenants et aboutissants de la gestion du partage des ressources ;
7. Comprendre les différents liens entre le mémoire physique, logique et virtuelle ;
8. Utiliser et implanter certains services et fonctionnalités du système d'exploitation ;
9. Comprendre la mise en œuvre des différents algorithmes et politiques reliés aux systèmes d'exploitation dans des cas réels.

1.3 Contenu détaillé

Thème	Contenu	Heures
1	Introduction	1
2	Pilotes Concepts de pilotes sous Linux Écriture d'un pilote	4
3	Historique et types Présentation des différents types de systèmes d'exploitation de façon historique	1
4	Fonctions et services Définition et présentation des services et fonctions des systèmes d'exploitation	2
5	Systèmes de fichiers (a) Théorie - Notion de fichier, supports, organisation des fichiers - Structure et organisation des répertoires - Gestion de l'espace libre, allocation de l'espace, techniques de liaison - Protection (b) Études de cas - Unix, Venix, HP-UX, Linux, Primos, RT-11, MacOS, DOS, Windows (NTFS)	10
6	Gestion de l'UCT (a) Théorie - Concepts et composants (processus), niveaux d'ordonnement - Algorithmes d'ordonnement, critères d'évaluation - Algorithmes multiprocesseurs (b) Études de cas - Unix, Solaris, Linux, VAX-VMS, Windows, Primos	8
7	Gestion de la mémoire (a) Théorie - Techniques simples, <i>swapping</i> , partitions fixes et variables - Pagination, segmentation - Segmentation	8

8	Mémoire virtuelle (a) Théorie - Chevauchement, demande de pages et performance, concept de mémoire virtuelle - Algorithmes de remplacement de pages, algorithmes d'allocation - Écroulement, localité, autres considérations (b) Études de cas - VAX-VMS, Windows, Unix, Solaris, Linux, MVS (c) Nouvelles techniques de gestion de la mémoire	8
9	Interblocages (a) Exemples (b) Conditions d'interblocage et méthodes de traitement	2

2 Organisation

2.1 Méthode pédagogique

La matière du cours sera présentée lors de séances magistrales, aux horaires indiqués dans ce plan de cours. L'évaluation se fera à travers quatre (4) travaux pratiques, un examen intra et un examen final.

2.2 Calendrier du cours

	Semaine du	Thème
1	29-08-2016	1,2
2	05-08-2016	2,3,4
3	12-09-2016	5
4	19-09-2016	5
5	26-09-2016	5
6	03-10-2016	Révision
7	Période du 08-10-2016 au 15-10-2016	Examen périodique
8	17-10-2016	Relâche
9	24-10-2016	6
10	31-10-2016	6
11	07-11-2016	6,7
12	14-11-2016	7
13	21-11-2016	7,8
14	28-11-2016	8
15	05-12-2016	8
16	12-12-2016	Révision
17	Période du 13-12-2016 au 23-12-2016	Examen final

* Des séances spéciales auront lieu (dates et local à déterminer).

2.3 Évaluation

Travaux pratiques (4 x 10 %) : 40 %

Examen périodique : 20 %

Examen final : 40 %

2.4 Plagiat

Un document dont le texte et la structure se rapporte à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe, une attention spéciale sera portée au plagiat, défini dans le Règlement des études comme « le fait, dans une activité pédagogique évaluée, de faire passer indûment pour siens des passages ou des idées tirés de l'œuvre d'autrui. ». Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du Règlement des

études : « tout acte ou manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique. » À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera référé au Secrétaire de la Faculté des sciences.

2.5 Échéancier des travaux

TP	Réception	Thème	Remise
1	05-08-2016	Pilote sous Linux	14-09-2016
2	14-09-2016	Système de fichiers	08-10-2016
3	24-10-2016	Gestion de l'UCT	22-11-2016
4	22-11-2016	Mémoire virtuelle	13-12-2016

Directives particulières

Le premier travail consistera à écrire un pilote simple pour le système d'exploitation Linux, afin de comprendre comment un système d'exploitation s'interface avec le matériel. Tous les travaux suivants portent sur le système Nachos. Nachos est un mini-système d'exploitation auquel il manque des morceaux. Vos travaux consisteront à compléter le système.

Ces travaux seront à remettre à l'aide du programme *turnin*. Une date limite de remise vous sera rappelée lors de la distribution des énoncés; tout dépassement de cette date entraînera automatiquement la note zéro.

Les travaux seront à faire sous Unix et seront testés sur la machine *tarin*. Un non-fonctionnement dans cet environnement entraînera également la note zéro.

Une bonne qualité de code et de documentation est attendue. Il est fortement recommandé de respecter les normes départementales de programmation. Pour mémoire, ces normes sont disponibles à cette adresse:

<http://www.dmi.usherb.ca/~fraikin/cours/Normes/normes-de-programmation.pdf>

Toute situation éventuelle de plagiat se verrait traitée selon les normes du «Règlement des études» de l'Université de Sherbrooke, disponible à cette adresse:

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

Des consignes supplémentaires ou des modifications pourront vous être communiquées au cours de la session; vous serez tenus de les respecter.

3 Matériel nécessaire pour le cours

SILBERSCHATZ (A.), GALVIN (P.B.) et GAGNE (G.), Operating Systems Concepts (8th Edition), John Wiley, 2009 (*disponible à la Coopérative*)

4 Références

- [1] BACH (M.J.), The Design of the Unix Operating System, Prentice-Hall Software Series, Prentice-Hall, 1986.
- [2] BACON (J.) et HARRIS (T.) Operating System : Concurrent and Distributed Software Design, Addison-Wesley, 2003.
- [3] BIC (L.F.) et SHAW (A.C.), Operating System Principles, Prentice-Hall, 2003.
- [4] CHOW (R.) et JOHNSON (T.), Distributed Operating System and Algorithms, Addison-Wesley, 1997.

- [5] DEITEL (H.M.), DEITEL (P.J.) et CHOFFNES (D.R.), Operating System, Prentice-Hall, 2004.
- [6] LYNCH (N), Distributed Algorithms, Morgan Kaufmann Publishers, 1996.
- [7] NUTT (G.), Operating Systems, Addison-Wesley, 2003.
- [8] RUBINI (B.), LINUX device Drivers, O'Reilly, 1998 Prentice-Hall, 2003.
- [9] SCHNEIER (B.), Applied Cryptography, John-Wiley, 1996 Prentice-Hall, 2003.
- [10] STALLINGS (W.), Operating Systems : Internals and Design Principles (Fourth Edition), Prentice-Hall, 2001.
- [12] TANENBAUM (A.S.), Modern Operating Systems, Prentice-Hall, 2001.
- [13] TANENBAUM (A.S.) et VAN STEEN (M), Distributed Systems : Principles and Paradigms, Prentice-Hall,2001.