

## Systèmes d'exploitation

<b>Domaine</b>	Ingénierie et Architecture
<b>Filière</b>	Informatique et systèmes de communication
<b>Orientation</b>	Sécurité informatique (ISCS)
<b>Mode de formation</b>	Temps partiel/En emploi

### Informations générales

Nom	: Systèmes d'exploitation
Identifiant	: SYE
Année académique	: 2022-2023
Responsable	: Daniel Rossier
Charge de travail	: 90 heures d'études
Périodes encadrées	: 64 (= 48 heures)

Semestre	E1	S1	S2	E2	S3	S4	E3	S5	S6	E4	S7	S8
Cours								32				
Laboratoire								32				

### Connaissances préalables recommandées

L'étudiant-e doit connaître et savoir utiliser les notions suivantes :

- un langage de programmation système tel que le langage C (notions de pointeur, allocation mémoire, etc.) ;
- le fonctionnement d'un système à processeur dans les grandes lignes ;
- les structures de données (listes, tableaux, piles, etc.) ;
- les algorithmes de base (tri, parcours de liste, fonctions récursives, etc.).

Les unités d'enseignement PRG2 (programmation), ASD (algorithmes et structures de données) et ARO (architecture des ordinateurs) permettent d'acquérir ces connaissances.

### Objectifs

Le développement d'un nouveau système informatique conduit très souvent à l'adaptation de différents composants logiciels tels que moniteurs (ou *bootloaders*), systèmes d'exploitation (Linux, Android, RTOS, etc.), pilotes de périphériques (*drivers*), etc.

C'est pourquoi, l'ingénieur-e en informatique doit maîtriser les aspects liés au code de bas niveau utilisé notamment lors du démarrage d'un OS (initialisation des *timers*, de la mémoire, des périphériques, etc.) afin d'être ensuite capable de l'adapter sur différentes plates-formes.

A l'issue de cette unité d'enseignement, l'étudiant-e sera capable de :

- comprendre les différentes architectures d'un OS et les interactions avec le matériel ;
- comprendre la notion d'espace d'adressage ;
- comprendre la notion d'appel système ;
- concevoir une application avec des appels système de base ;
- comprendre la notion de processus et de *threads* ;
- comprendre la notion d'IPC ;
- concevoir et réaliser une application faisant intervenir des IPCs ;

- décrire le fonction d'un ordonnanceur et ses différentes politiques ;
- comprendre les mécanismes de gestion d'adressage virtuel ;
- comprendre les différents algorithmes d'allocation mémoire ;
- analyser un système de pagination à un et deux niveaux ;
- comprendre les mécanismes et algorithmes de remplacement de pages physiques ;
- comprendre le fonctionnement d'un système de fichiers dans un OS ;
- connaître certaines parties de code de noyau et d'applications système dans l'environnement SO3 dont les principes sont très proches de ceux de *Linux*.

## Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

**Cours:** 32 périodes

- Caractéristiques d'un OS, introduction à SO3, environnement & IDE, structure des projets, mémoire & espaces d'adressage 2
- Architecture matérielle, modes d'exécution du CPU, notions d'interruption, architectures des systèmes d'exploitation 2
- Appels systèmes, construction d'une image binaire, définition d'un processus 2
- Structure PCB (Process Control Block), création de processus, appels systèmes fork(), exec(), waitpid(), exit(), changement de contexte 2
- Etats et transitions, définition d'un thread, librairie de threads POSIX, exécution de threads 2
- Descripteurs de fichier (file descriptor), sous-système IPC (Inter-Process Communication), introduction aux tube (pipe), tubes nommés et anonymes 2
- Introduction aux signaux, introduction aux sockets réseau, appels systèmes liés aux sockets, approche client-serveur 2
- Ordonnancement et mesures, notion de préemption et points de préemption, ordonnancement FCFS, SJF, RR, ordonnancement par priorité, ordonnancement par files d'attentes multiples 2
- Dispositifs mémoire, allocation de la mémoire physique, algorithme d'allocation First-Fit, Best-Fit, Quick-Fit, espace d'adressage virtuel 2
- Introduction à la pagination, pagination à un niveau, notion de TLB (Translation Look-aside Buffer), pagination multiniveaux 2
- Gestion des pages physiques, extension de la mémoire, concept de faute de page 2
- Notion de remplacement de page, algorithmes OPT, FIFO, LRU, algorithme de la seconde chance, algorithme WSClock 2
- Organisation d'un stockage secondaire, système de fichiers et partitions, allocation contiguë, allocation par listes chaînées 2
- Allocation indexée, notion de liens durs et symboliques 2
- Travaux écrits 4

**Laboratoire:** 32 périodes

- Première incursion avec le debugger dans Eclipse, utilisation du debugger pour une application (espace utilisateur), modification d'octet (byte) en mémoire, modification du shell, modification de la version du noyau, modification de la routine de service (ISR) du timer 4
- Rajout d'un appel système dans SO3, rajout d'une commande de parallélisation dans le shell 4
- Création et exécution de threads, communication client-serveur au travers de tubes (pipes) 4
- Application client-serveur TCP, utilisation de signaux 4
- Ajout du l'appel système « renice() » et gestion des priorité, exploration de la gestion mémoire dans le noyau de SO3 4
- Implémentation d'un algorithme de traduction d'adresse, ajout de bits d'attribut dans les PTEs, changement de mémoire 4
- Implémentation de l'algorithme LRU, évolution vers l'algorithme WSClock 4

- Amélioration de l'application "Is", création de liens symboliques avec FAT-32

4

### Bibliographie

Aucune information

### Contrôle de connaissances

**Cours** : l'acquisition des matières de cet enseignement sera contrôlée au fur et à mesure par des tests et des travaux personnels tout au long de son déroulement. Il y aura au moins 2 tests d'une durée totale d'au moins 2 périodes.

**Laboratoire** : ils seront évalués sur la base des rapports de manipulation, à 3 reprises au minimum.

**Examen** : L'atteinte de l'ensemble des objectifs de formation sera vérifiée lors d'un contrôle final commun écrit d'une durée de 90 minutes.

Matériel autorisé :

- Information communiquée directement par l'enseignant.

### Calcul de la note finale

Note finale = moyenne cours x 0.3 + moyenne laboratoire x 0.2 + moyenne examen x 0.5