

# Systemes d'exploitation

## Syllabus

### Enseignant responsable

Thibaud Martinez

[thibaud.martinez@dauphine.psl.eu](mailto:thibaud.martinez@dauphine.psl.eu)

### Présentation

Ce cours est une introduction aux concepts des systèmes d'exploitation pour les étudiants en **troisième année de licence informatique à l'université Paris-Dauphine**. Il s'intéresse au fonctionnement des systèmes Unix, ceux-ci étant très répandus et existant dans des versions *open source*. Le cours conjugue études théoriques, au travers de cours magistraux, et travaux pratiques permettant d'illustrer la théorie.

### Organisation

- 15 heures de CM
- 15 heures de TP

### Prérequis

- Une précédente expérience avec un langage de programmation (C, C++, Java, etc.).
- Une expérience élémentaire avec l'utilisation d'un système d'exploitation (Unix de préférence).

### Objectifs du cours

1. Comprendre le rôle d'un système d'exploitation et de ses composants.
2. Être capable d'interagir avec un système d'exploitation de type Unix grâce à l'interpréteur de commandes.
3. Comprendre le lien entre développement applicatif et système d'exploitation.
4. Maîtriser les bases de la programmation système sous Unix en C.
5. Comprendre les mécanismes centraux d'un système d'exploitation (virtualisation, concurrence et persistance) et leurs implémentations.

## Évaluation

- Projet informatique (30 % de la note finale).
- Examen terminal (70 % de la note finale).

## Ouvrages de référence

Ce cours est basé sur l'excellent **Operating Systems: Three Easy Pieces** de *Remzi H. Arpaci-Dusseau* et *Andrea C. Arpaci-Dusseau*. L'ouvrage est disponible librement sur [son site Internet](#).

Il est nécessaire d'acquérir une maîtrise basique du langage de programmation C pour ce cours. Des rappels du langage sont donnés dans le cours.

L'ouvrage de référence recommandé pour le langage C est **Effective C** par *Robert C. Seacord*.

Le classique **The C Programming Language, 2nd Edition** par *Brian W. Kernighan* et *Dennis M. Ritchie* vaut toujours le détour même s'il ne tient pas compte des développements les plus récents du langage.

Pour une vue d'ensemble relativement courte des éléments clés du langage C (45 pages), on pourra se référer au document **Essential C** mis gracieusement à disposition par [Stanford CS Education Library](#).

## Programme

Date	Type	Titre	Chapitres
3/02	CM	Ch 1 - Introduction	<a href="#">Introduction</a>
13/02	TP	TP 1 - Utilisation du shell	
16/02	CM	Ch 2 - Virtualisation (CPU) : les processus	<a href="#">Processes</a> • <a href="#">Process API</a>
6/03	TP	TP 2 - Processus	
8/03	CM	Ch 3 - Virtualisation (CPU) : ordonnancement	<a href="#">Limited Direct Execution</a> • <a href="#">CPU Scheduling</a> • <a href="#">Multi-level Feedback</a>
9/03	TP	TP 2 - Processus (suite)	
20/03	CM	Ch 4 - Virtualisation (mémoire) : segmentation	<a href="#">Address Spaces</a> • <a href="#">Memory API</a> • <a href="#">Address Translation</a> • <a href="#">Segmentation</a>
22/03	TP	TP 3 - Ordonnancement	
3/04	CM	Ch 5 - Virtualisation (mémoire) : pagination	<a href="#">Paging</a> • <a href="#">Faster Translations (TLBs)</a> • <a href="#">Smaller Tables</a>
5/04	TP	TP 4 - Mémoire	
19/04	CM	Ch 6 - Concurrency : threads et verrous	<a href="#">Concurrency</a> • <a href="#">Thread API</a> • <a href="#">Locks</a>
20/04	TP	Projet	
10/05	CM	Ch 7 - Concurrency : condition variables et sémaphores	<a href="#">Condition Variables</a> • <a href="#">Semaphores</a> • <a href="#">Common Concurrency Problems</a>
11/05	TP	TP 5 - Threads et verrous	

Date	Type	Titre	Chapitres
24/05	CM	Ch 8 - Persistence : périphériques d'entrées-sorties	<a href="#">I/O Devices</a> • <a href="#">Hard Disk Drives</a>
25/05	TP	TP 6 - Sémaphores	
7/06	CM	Ch 9 - Persistence : implémentation des systèmes de fichiers	<a href="#">Files and Directories</a> • <a href="#">File System Implementation</a>
7/06	CM	Ch 10 - Persistence : cohérence du système de fichiers et journalisation	<a href="#">FSCK and Journaling</a>
8/06	TP	TP 7 - Interface du système de fichiers	
21/06	Examen	Examen	
22/06	CM/TP	Ch 11 - Une introduction à Docker et aux conteneurs	