

Portage des systèmes d'exploitation

Domaine	Ingénierie et Architecture
Filière	Informatique
Orientation	Informatique embarquée (IE)
Mode de formation	Temps partiel/En emploi

Informations générales

Nom:	:	Portage des systèmes d'exploitation
Identifiant:	:	POS
Années académiques	:	2017-2018, 2018-2019, 2019-2020
Responsable:	:	Daniel Rossier
Charge de travail:	:	120 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 doit être à l'aise avec la programmation C de bas niveau, ainsi qu'avec les notions de systèmes d'exploitation. En particulier, il doit connaître l'architecture classique d'un OS monolithique.

Objectifs

Le développement d'un nouveau système embarqué 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 en informatique embarquée doit maîtriser les aspects liés au code de bas niveau utilisé notamment lors du démarrage d'un OS sur un système embarqué (initialisation des timers, de la mémoire, des périphériques, etc.). Il sera capable ensuite de l'adapter sur différentes plates-formes.

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

- Identifier les différentes parties de code d'un OS (bootstrap, traitements bas niveau, ...)
- Comprendre la notion de HAL (Hardware Abstraction Layer) et de gestion CPU (cache, co-processeur, etc)
- Analyser le gestionnaire d'un timer système
- Analyser le gestionnaire d'un système d'interruptions vectorisées (niveau IRQ)
- Analyser les fonctions d'accès mémoire

A l'issue des travaux pratiques en laboratoire, principalement destinés à l'assimilation des connaissances et à l'acquisition d'expérience, l'étudiant-e sera en outre capable de

- Porter un code de bootstrap sur un CPU spécifique pour une plate-forme choisie
- Porter un code de gestion mémoire sur un CPU spécifique et pour une plate-forme choisie

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- Amorçage d'un noyau (bootstrap), moniteur, initialisation bas niveau, relocation de code, code générique/architecture-dépendant 8
- Gestion du matériel à l'aide de modèles abstraits (HAL), accès matériel CPU et coprocesseur, accès aux antémémoires (caches) 6
- Rôle d'un timer système, techniques de gestion (périodique/oneshot) 4
- Mécanismes d'interruption IRQ, gestion hiérarchisée des IRQs 6
- Mécanismes de gestion mémoire de bas niveau, accès MMU, gestion des espaces d'adressage virtuelles, gestion des descripteur de la mémoire physique 8

Laboratoire: 32 périodes

- Portage sur un CPU ARM de portions de code spécifique au bootstrap d'un OS 10
- Portage sur un CPU ARM de portions de code spécifique à la gestion du timer système, du gestionnaire d'IRQs et du gestionnaire mémoire 22

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 3 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.

Calcul de la note finale

Note finale = moyenne cours x 0.6 + moyenne laboratoire x 0.4

Fiche validée le 18.08.2017 par Donini Pier