

Programmation temps réel

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

Informations générales

| | | |
|---------------------|---|---|
| Nom: | : | Programmation temps réel |
| Identifiant: | : | PTR |
| Années académiques | : | 2015-2016, 2016-2017, 2017-2018, 2018-2019, 2019-2020 |
| Responsable: | : | Yann Thoma |
| 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 connaître et savoir utiliser les notions suivantes :

- bases du langage C (structures, types, manipulation des pointeurs) ;
- fonctionnement général d'un processeur, en particulier les interruptions.

Les unités d'enseignement APR1-2 ou INF2 (informatique, langage C), et MUI (microinformatique) ou ASP (architecture des systèmes à processeur) permettent d'acquérir ces connaissances.

Conditions pour la programmation automatique de cette unité :

- L'étudiant-e doit avoir obtenu une note supérieure ou égale à la limite de compensation dans les unités : APR1, APR2, INF2;
- L'étudiant-e doit avoir suivi ou suivre en parallèle les unités : MUI, ASP.

Objectifs

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

- décrire l'architecture d'un système d'exploitation temps-réel ;
- décrire les interactions entre le matériel et le logiciel, notamment par le concept d'interruption et de tâche ;
- déterminer les causes du temps de latence et de la gigue dans un système temps-réel ;
- choisir une politique d'ordonnancement adéquate en tenant compte des contraintes temporelles ;
- énoncer les critères d'ordonnancabilité liés à une politique d'ordonnancement ;
- déterminer le temps de réponse d'une application temps-réel ;
- examiner une API d'un noyau temps-réel afin de déterminer l'utilisation des différents services et fonctions disponibles ;

- d'une manière générale, développer une application temps-réel d'une complexité moyenne en utilisant les différents services d'un noyau temps-réel ;
- exploiter les différents mécanismes de communication et de synchronisation temps-réel en adéquation avec la complexité de l'application ;
- déterminer les situations d'inversion de priorité et expliquer comment y remédier ;
- identifier les situations d'inter-blocages et expliquer les mécanismes afin de les éviter.

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

- utiliser un outil de simulation d'ordonnement temps-réel en comprenant les différents paramètres et résultats possibles ;
- implémenter un traitement spécifique en réponse à une interruption matériel dans une application composée de tâches temps-réels ;
- réaliser une application temps-réel (strict) d'une certaine complexité en utilisant les services offerts par un noyau temps-réel.

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- Introduction au temps-réel : définitions et classification du temps réel, architecture d'un exécutif temps-réel, séparation des espaces "utilisateur" et "noyau", notion de temps et contraintes temporelles, langage de programmation temps-réel, types de tâches (périodiques, apériodiques et sporadiques), service de gestion de tâches. 6
- Interactions matériel/logiciel : rappel sur les interruptions, routines de service (ISR), notion de temps de latence et gigue, priorités masquage et tâches différées, service de gestion du temps (timer, watchdogs, etc.). 4
- Techniques d'ordonnement : introduction aux techniques d'ordonnement (systèmes préemptifs/non-préemptifs), ordonnancement coopératif, notion d'ordonnabilité, politiques RM, DM et EDF, calcul du temps de réponse, traitement des tâches apériodiques, tâches dépendantes. 10
- Communication et synchronisation inter-tâche : service de gestion mémoire, communication par file de message, synchronisation par drapeau événement, notion de sections critiques, services de mutex et de sémaphore, variables condition, problème de l'inversion de priorité, notion d'inter-blocage et approche des solutions. 12

Laboratoire: 32 périodes

- Simulation de différents ordonnancements temps-réel. 4
- Introduction à un exécutif temps-réel basé sur l'extension Xenomai/Linux. 4
- Utilisation des services de tâches et d'interruption. 6
- Réalisation d'une application temps-réel faisant intervenir de la communication et de la synchronisation entre les tâches. 18

Bibliographie

<http://reds.heig-vd.ch/formations/bachelor/PTR/PTRdoc.aspx>

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

Fiche validée le 30.06.2015 par Donini Pier