

Programmation assembleur

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

Informations générales

Nom:	:	Programmation assembleur
Identifiant:	:	ASM
Années académiques	:	2016-2017, 2017-2018, 2018-2019, 2019-2020
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 doit connaître un modèle abstrait d'un processeur.

Objectifs

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

- Comprendre les notions d'environnement croisé, de BSP, de datasheet
- Comprendre les environnements matériels de mise au point (JTAG)
- Comprendre les notions d'assembleur et caractéristiques d'un processeur
- Comprendre le jeu d'instructions d'un processeur, incluant les instructions de traitement, de transferts de données et de branchement
- Comprendre la notion de mode d'adressage
- Connaître les extensions du langage et directives de compilation
- Comprendre les appels de fonction, conventions d'appel, et gestion de la pile
- Décrire le rôle d'un compilateur et du code généré (code objet)
- Expliquer les mécanismes d'interactions C-assembleur avec l'assembleur inline et les différentes directives (entrées, sorties, clobbers, etc.)
- Concevoir le code en assembleur inline dans une code C pour le passage de structures conventionnelles (valeurs, adresses)

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

- Maîtriser un environnement de développement croisé avec une cible à processeur ARM
- Implémenter différents code en assembleur inline effectuant des passages in/out de différentes valeurs
- Implémenter des passages de valeurs entre C et assembleur ARM et x86
- Concevoir et développer une petite application en assembleur ARM et/ou x86

Contenu et formes d'enseignement

Répartition des périodes indiquée à titre informatif.

Cours: 32 périodes

- Environnement de développement croisé, BSP	2
- Notion de JTAG, appliquée au debug du processeur	1
- Langage d'assemblage, contenu d'un fichier source	4
- Modes d'exécution sur un processeur de type ARM et de type Intel x86	1
- Jeux d'instructions ARM/x86, instructions de traitement, de transferts de données et de branchement	8
- Modes d'adressage	2
- Pseudo-instructions, directives de compilation, macro-instructions	6
- Fonctions, conventions d'appel, gestion de la pile	6
- Code objet, traduction C-assembleur	2

Laboratoire: 32 périodes

- Monitoring d'une application ARM tournant sur cible émulée	4
- Monitoring d'une application ARM tournant sur cible réelle	4
- Investigations sur processeur ARM et co-processeurs associés	4
- Conception et réalisation d'une application en ARM	6
- Conception et réalisation d'une application en Intel/x86	6
- Présentation de l'assembleur inline (gcc) avec les différentes directives (entrées, sorties, liste de clobbers)	4
- Conception et implémentation de passages de valeurs et d'adresses en C-assembleur ARM	2
- Conception et implémentation de passages de valeurs et d'adresses en C-assembleur x86	2

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

Fiche validée le 17.08.2016 par Sanchez Eduardo