

Compilation :

De l'algorithme à la porte logique

Cours 2eme année, voie d'ouverture, domaine informatique., 45 heures en 15 séances de Cours Intégrés

Objectifs :

Étudier l'enchaînement menant de l'écriture d'un algorithme dans un langage de programmation à son exécution par un ordinateur ;
Comprendre les différents outils conceptuels et techniques mis en jeu dans l'écriture d'un compilateur (ou interpreteur, ou traducteur) ;
Mettre en œuvre ces outils à travers la réalisation d'un compilateur d'un sous-ensemble de Java vers l'assembleur Mips.

Cet objectif général du cours se décline sur 3 composantes plus spécifiques :

- Étudier l'architecture des processeurs et l'écriture de code assembleur
- Introduire la théorie des Langages et Maitriser ses utilisations pratiques en informatique à travers les outils de génération de compilateur (*compiler's compiler*).
- Approfondir la maitrise de la programmation dans un langage évolué. D'une part, par la compréhension de la façon dont les paradigmes de programmation sont implantés pour leurs exécutions. D'une part, par la pratique dans l'écriture (en Java) d'un compilateur,;

Contenu :

Analyse Syntaxique

- Chaîne de compilation, analyse lexicale et syntaxique, hiérarchie de Chomsky
- Expression Régulières et utilisation de Jflex
- Automate fini et Langage régulier
- Grammaire algébrique et utilisation de Cup
- Analyse syntaxique LR

Architecture de processeur

- De la porte logique au processeur
- Instructions, registres, assembleur (Mips)
- Mémoire, appel de fonction, pile, frame, syscall

Compilation et écriture d'un compilateur

- Parcours d'arbres et Pattern Visitor
- Analyse sémantique, table de symbole, contrôle de type
- Génération de code intermédiaire
- Génération de code, modèle d'appel, allocation de registre.

Mots clés :

analyse lexicale, analyse syntaxique, analyse sémantique, génération de code, processeur, registre, pile et tas, assembleur(Mips)
expressions régulières, langage algébrique, *compiler's compiler* (jflex,cup)
Arbre de Syntaxe Abstraite, table de symbole, contrôle de type, code intermédiaire.
Patron de conception Visiteur.

Hors champs :

Les éléments suivants ne font pas partie du cours, ou uniquement de façon marginale :

- aspects théoriques de l'analyse sémantique : grammaires attribuées
- analyse sémantique dynamique et analyse de flow
- transformation d'arbres et techniques d'optimisation de code
- optimisation algorithmique et structures de données dans l'écriture d'un compilateur.
- compilation séparée et édition de lien

Notation :

1 TP/DM Noté à mi-parcours (40%)
1 Rendu final sur le projet fil rouge du cours (60%)

Prérequis : Algorithmique et programmation en Java.

Responsables : Pascal Hennequin, Gaël Thomas