Compilation :
Du langage de haut niveau à l'assembleur

Module d'ouverture de 2ème année, domaine informatique, 45 heures en 15 séances de Cours Intégrés avec TP.

Objectifs :

1. Étudier l'enchaînement menant de l'écriture d'un programme dans un langage de haut niveau à son exécution.
2. Comprendre les différents outils conceptuels et techniques mis en jeu dans la réalisation d'un compilateur (ou interpréteur ou traducteur).
3. Mettre en œuvre ces outils à travers la réalisation d'un compilateur d'un sous-ensemble de JAVA vers l'assembleur MIPS.

Ces objectifs généraux du cours se déclinent sur trois directions :

- étudier l'architecture des processeurs et l'écriture de code assembleur ;
- introduire la théorie des langages et maîtriser ses utilisations pratiques en informatique à travers les outils de génération de compilateur (*compiler-compiler*) ;
- approfondir la maîtrise de la programmation dans un langage évolué, d'une part, par la compréhension de la façon dont les patrons de conception sont mis en œuvre, et d'autre 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 l'outil JFlex,
- automate fini et langage régulier,
- grammaire algébrique et utilisation de l'outil CUP,
- analyse syntaxique LR.

Architecture de processeur :
- structure d'un processeur,
- instructions, registres, assembleur (MIPS),
- mémoire, appel de fonction, pile, cadre d'appel, appel système.

Compilation et réalisation d'un compilateur :
- parcours d'arbres et patron de conception Visiteur,
- 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),
expression régulière, langage algébrique, *compiler-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 flux,
- transformation d'arbres et techniques d'optimisation de code,
- optimisation algorithmique et structures de données dans la réalisation d'un compilateur,
- compilation séparée et édition de lien.

Notation :

2 TP notés à mi-parcours et en fin de module (50%)
1 Rendu final sur le projet fil rouge du cours (50%)

Prérequis : algorithmique et structures de données, et programmation en JAVA

Responsables : Denis Conan et J. Paul Gibson