

Projet de compilation - Modules PCL1 et PCL2

L'objectif de ce projet est d'écrire un compilateur d'un langage dit de "haut niveau", en développant toutes les étapes depuis l'analyse lexicale jusqu'à la production de code assembleur. Ce langage que nous dénommerons CHAOS¹ a été décrit par A. Appel (Princeton University). Votre compilateur produira en sortie du code assembleur ARM, code assembleur que vous avez étudié dans le cours d'ASM de première année.

Le sujet ci-dessous décrit l'organisation des modules PCL1-PCL2 ainsi que le travail à réaliser.

1 Réalisation du projet

L'écriture du compilateur complet concerne les deux modules PCL1 et PCL2. La partie PCL1 (d'octobre à mi-janvier) s'adressera à tous les élèves : il s'agit d'écrire la grammaire du langage, puis de mettre en oeuvre les analyses lexicale, syntaxique et sémantique. La partie PCL2 (de février à mi-avril) concernera uniquement la génération de code assembleur. Seuls les élèves des approfondissements IL et ISS sont concernés par ce second module.

Pour ce projet, les élèves des approfondissements IAMD, SLE et SIE formeront des groupes de 3, tandis que les élèves des autres approfondissements travailleront par groupes de 4.

Les 18 élèves en mobilité au semestre 8 intégreront des groupes formés avec les élèves des approfondissements IAMD, SLE et SIE, puisqu'ils ne sont concernés que par la partie PCL1.

Vous utiliserez l'outil ANTLR-4, générateur d'analyseurs lexical et syntaxique *descendant*, interfacé avec le langage Java pour les étapes d'analyse lexicale et syntaxique. Vous générerez ensuite le code assembleur au format ARM. Votre compilateur doit signaler les erreurs lexicales, syntaxiques et sémantiques rencontrées. Lorsqu'une de ces erreurs lexicales ou sémantique est rencontrée, elle doit être signalée par un message explicite comprenant, dans la mesure du possible, un numéro de ligne. Votre compilateur peut s'arrêter après chaque erreur syntaxique détectée (pas d'obligation de reprise). En revanche, votre compilateur doit impérativement poursuivre l'analyse après avoir signalé une erreur sémantique.

Vous utiliserez un dépôt Git sur le Gitlab de TELECOM Nancy. Vous devrez créer votre projet dans votre espace personnel, il devra être privé et l'identifiant sera de la forme login1 (où login1 est le login du membre chef de projet de votre groupe). Vous ajouterez Suzanne Collin et Sébastien Da Silva en tant que "Master" de votre projet. Votre répertoire devra contenir tous les fichiers sources de votre projet, les dossiers intermédiaires et finals (au format PDF) ainsi qu'un *mode d'emploi* pour utiliser votre compilateur.

Les dépôts seront régulièrement consultés par les enseignants chargés de vous évaluer lors des séances de TP et lors de la soutenance finale de votre projet.

En cas de litige sur la participation active de chacun des membres du groupe au projet, le contenu de votre projet sur le dépôt sera examiné. Les notes peuvent être individualisées.

2 Première partie : module PCL1

PCL1 : séances TP 1 à 4 - Prise en main du logiciel ANTLR, définition complète de la grammaire du langage, de l'AST et de la TDS, réalisation des contrôles sémantiques.

On vous propose une initiation au logiciel ANTLR lors des deux premières séances de TP des semaines 41 et 42. Vous définirez ensuite la grammaire du langage et la soumettrez à ANTLR afin qu'il génère l'analyseur syntaxique descendant. Bien sûr, l'étape d'analyse lexicale est réalisée parallèlement à l'analyse syntaxique.

Vous aurez testé votre grammaire sur des exemples variés de programmes écrits en langage CHAOS, avec et sans erreurs lexicales et syntaxiques. Une évaluation de cette partie aura lieu lors des TP de la semaine 46.

¹CHAOS : Compilateur Hautement Algorithmique et Open Source - Christophe Bouthier...

Vous réfléchirez ensuite à la construction de l'arbre abstrait et de la table des symboles. Vous implémenterez ensuite tous les contrôles sémantiques liés à ce langage. Vous aurez réfléchi à la structure de la TDS et commencé les contrôles sémantiques pour la seconde évaluation qui aura lieu la semaine 49.

Lors des TP des semaines 46 et 49 nous noterons l'avancement de vos projets. Ces "évaluations" entreront dans la note finale du module PCL1.

La soutenance finale de cette partie PCL1 est fixée au 11 janvier 2023.

Vous montrerez lors de cette soutenance l'arbre abstrait, la table des symboles (une visualisation, même sommaire, est indispensable) ainsi que l'ensemble des contrôles sémantiques implémentés.

Il est impératif que vous ayez prévu pour la soutenance des exemples de programmes permettant de tester votre projet. Ces exemples ne seront pas à écrire le jour de la démonstration.

Vous rendrez au cours de ce module des **rapports d'activité** (max. 4 pages) présentant le travail réalisé et les difficultés rencontrées. Vous pouvez inclure dans ces documents la grammaire du langage, la structure de l'arbre abstrait et de la structure de la table des symboles, quelques jeux d'essais, etc. N'oubliez pas les éléments de gestion de projet (répartition du travail et des tâches au sein du groupe par exemple) pour le dernier rapport d'activité.

Vous remettrez ces rapports d'activité à votre enseignant de TP lors des séances PCL1 des semaines 46 et 49. Le dernier rapport sera à déposer dans son casier le lundi 9 janvier à 16h

3 Seconde partie : module PCL2

Génération de code assembleur ARM

Vous continuez votre projet par l'étape de génération de code.

Pour cette dernière phase, vous veillerez à générer le code assembleur de manière incrémentale, en commençant par les structures "simples" du langage.

Le code généré devra être en langage d'assemblage ARM, langage étudié dans le module *ASM* de première année.

Des séances de TP sont prévues pour cette seconde partie PCL2. Nous reviendrons vers vous pour les consignes.

A la fin du projet, donc à la fin du module PCL2, et pour le **mercredi 12 avril 2023 - 18h**, vous rendrez un dossier qui complètera les précédents rapports d'activités de PCL1 et comprendra *au moins* :

- les schémas de traduction du langage proposé vers le langage assembleur
- des *jeux d'essais* mettant en évidence le bon fonctionnement de votre compilateur, et ses limites éventuelles.
- une partie *gestion de projet*, à savoir une fiche d'évaluation de la répartition du travail sur cette seconde période avec la répartition des tâches au sein de votre binôme, l'estimation du temps passé sur chaque partie du projet, et le Gantt final.
- les divers CR de réunion que vous avez rédigés.

La fin du projet est fixée au jeudi 13 avril 2023, date prévue pour les soutenances finales.

Lors de cette soutenance, on vous demandera de faire une démonstration de votre compilateur. Un planning vous sera proposé pour fixer l'ordre de passage des groupes.

Il est impératif que vous ayez prévu des exemples de programmes permettant de tester votre projet et ses limites (ces exemples ne seront pas à écrire le jour de la démonstration) : vous nous montrerez lors de votre soutenance votre "plus beau" programme écrit en langage CHAOS...

Aucun délai supplémentaire ne sera accordé pour la fin du projet. Le temps restant sur les mois d'avril et de mai est réservé à la finalisation de vos autres projets et du PIDR.

Concernant l'évaluation de la génération de code, on vous fournira 4 "niveaux" de programmes écrits dans le langage du projet, ce qui vous permettra de vous situer dans cette phase de génération de code. Le niveau 2 est

généralement celui requis pour obtenir une note de 10/20. Ceci n'est qu'une indication, car la note finale dépend aussi des évaluations intermédiaires et du dossier rendu.

Bien entendu, il est interdit de "s'inspirer trop fortement" du code d'un autre groupe; vous pouvez discuter entre-vous sur les structures de données à mettre en place, sur certains points techniques à mettre en oeuvre, etc... mais il est interdit de copier du code source sur vos camarades. Si cela devait se produire, nous saurons en tenir compte dans nos évaluations.

4 Présentation du langage

Comme il est inutile de retranscrire le manuel de référence du langage CHAOS édité par les auteurs de Tiger, une version du manuel de référence est fournie avec le sujet et vous servira pour définir la grammaire et effectuer les contrôles sémantiques.

Quelques simplifications sont cependant apportées sur les aspects lexicaux :

- les commentaires ne peuvent pas être imbriqués,
- les chaînes de caractères sont composées d'une suite non vide de caractères imprimables. Elles s'écrivent entre des guillemets (caractère ").

Concernant la section *Standard Library* du manuel, cette section sera très simplifiée et comportera uniquement la sortie suivante: **print(x)** qui écrit un entier sur une ligne de la sortie standard.

Le terme **print** est un mot-clé du langage.