

Sivatarsiban Steeve SIVANANTHAM

Damien Chesneau

Projet de compilation

Pour ce projet de compilation voici la structure que nous avons choisi :

- fichier .tpc (code source de notre compilateur)
- makefile
- module producer
- module symbols_table
- module vm_functions
- fichier tpc.y (Bison)
- fichier tpc.lex (Flex)

Grâce à cette répartition il a été plus simple de développer notre projet.

Le module producer

Dans ce module, nous avons regroupé toutes les fonctions que l'on devait appliquer à la grammaire. C'est la logique du compilateur. Pour plus de détails, voir producer.h

Le module symbols_table

Ce module est notre table des symboles. Il nous permet de garder en mémoire l'emplacement de chaque variable. Pour plus de détails, voir symbols_table.h

Le module vm_functions

Ce module a un fonctionnement très simple il est simplement chargé de bien formater les fonctions de la machine virtuelle. Pour plus de détails, voir vm_functions.h

Difficultés

Nous avons eu plusieurs difficultés pendant ce projet mais particulièrement celle-ci :

- Impossibilité de récupérer le type sur les variables globales: En effet, Bison ne sachant pas quel non-terminal choisir il y avait un conflit entre deux règles. Malheureusement nous n'avons pas eu le temps de remédier à ce problème et avons dû nous contraindre à gérer uniquement les variables de type entier en tant que variables globales. Bien entendu les caractères fonctionnent en tant que variables locales.
- L'implémentation des fonctions nous a aussi posé problème. Nous avons eu du mal à trouver un moyen d'échanger les informations entre les fonctions tout en gardant une certaine localité des fonctions. Nous avons mis en place cela via un `ld`

(différent du label) qui désigné une fonction et en ajoutant les éléments dans la table des symboles en fonction des cet Id.