Documentation Utilisateur

Compilateur IFCC (IF-C-Compiler)

Hexanome 4211 : DALAOUI Riad, AUBUT Antoine, CHIKHI Djalil, HANADER Rayan, TAIDER Sami, CHAOUKI Youssef, and LOUVET Samuel

INSA Lyon - Département Informatique - 4IF-PLD-COMP

Février - Avril 2025

Table des matières

1 Introduction

1.1 Présentation

Le compilateur `IFCC` permet de compiler des programmes écrits en `C` vers de l'assembleur `x86`.

1.2 Fonctionnalités principales

Le compilateur réalise l'analyse lexicale et syntaxique, la vérification sémantique ainsi que la génération de l'équivalent assembleur (`x86`) pour la liste de fonctionnalitées suivante :

- Un seul fichier source sans pré-processing. Les directives du pré-processeur sont autorisées par la grammaire, mais ignorées, ce afin de garantir que la compilation par un autre compilateur soit possible (exemple : inclusion de `stdio.h`)
- Les commentaires sont ignorés
- Type de données de base `int` (un type 32 bits) et `char` (avec simple quote)
- Variables
- Constantes
- Opérations arithmétiques de base : `+`, `-`, `*`
- Division et modulo
- Opérations logiques bit-à-bit : `|`, `&`, `^`
- Opérations de comparaison : `==`, `!=`, `<`, `>`
- Opérations unaires : `!` et `-`
- Déclaration de variables n'importe où
- Affectation (qui, en `C`, retourne aussi une valeur)
- Utilisation des fonctions standard `putchar` et `getchar` pour les entrées-sorties
- Définition de fonctions avec paramètres, et type de retour `int` ou `void`
- Vérification de la cohérence des appels de fonctions et leurs paramètres
- Structure de blocs grâce à `{` et `}`
- Support des portées de variables et du shadowing
- Les structures de contrôle `if`, `else`, `while`
- Support du `return` expression n'importe où
- Vérification qu'une variable utilisée a été déclarée
- Vérification qu'une variable déclarée est utilisée
- Possibilité d'initialiser une variable lors de sa déclaration
- Opérateur d'affectation += -= *=
- Incrémentation ++ et Decrementation –

La grammaire à employer pour rédiger le code qui sera accepté par le compilateur est la même que la grammaire `C` usuelle, dans la limite des fonction-

nalités énoncées ci-dessus.

2 Build et tests

2.1 Compilation

Les commandes pour compiler le compilateur `IFCC` sont les suivantes

```
cd pld-comp/compiler/
make clean
make
```

Elles présupposent que le fichier `/configs/config.mk` soit adapté à votre système d'exploitation, mais aussi et surtout à votre façon de télécharger ANTLR4 et son emplacement dans votre machine. N'oubliez donc pas **d'installer** ANTLR4 de la manière dont vous souhaitez, mais d'adapter ensuite les fichiers nécessaires.

2.2 Exécution des tests

```
cd ../tests
python3 ifcc-test.py [--verbose, --debug, --wrapper] testfiles/*.c
python3 ifcc-test.py [--verbose, --debug, --wrapper] new_tests/*.c
```

Cela présuppose que :

- Vous ayez un interpreteur Python d'installé et mis dans le PATH (si il est autre que `python3`, remplacez-le par la commande correcte).
- Les répertoires `pld-comp/tests/new_tests` et `pld-comp/tests/testfiles` ne contiennent que des fichiers à l'extension `.c`
- Que la compilation du compilateur `IFCC` se soit bien déroulée, et que le fichier `ifcc` soit présent dans `pld-comp/compiler`.
- Si vous ajoutez l'option de compilation `-wrapper` ou `-w`, vous ajoutiez à la suite le chemin vers le script shell que vous souhaitez utiliser à la place de celui par défaut (`ifcc-wrapper.sh`).

Si vous souhaitez ne compiler qu'un fichier `C`, par exemple : `exemple.c`, remplacez à la fin de la ligne de commande `testfiles/*.c` ou `new_tests/*.c` par `testfiles/exemple.c` ou `new_tests/exemple.c`

La compilation d'un nouveau fichier `C` n'est possible que depuis le repertoire /tests puisque c'est là que sont contenus les fichiers principalement utilisés pour cela : `ifcc-test.py` et `ifcc-wrapper.sh`. L'execution depuis un autre repertoire mènera à une erreur. Le fichier `C` peut, lui, être situé n'importe où dans la machine à condition que le bon chemin soit spécifié dans la commande de compilation.

Le projet est disponible sur GitHub au repository suivant :
https ://github.com/Samsam19191/C-Compiler