

Projet de compilation (ENSEIRB- Informatique, 2019-2020)

Introduction

Le projet de compilation a pour objet la réalisation d'un compilateur d'un mini langage appelé pour l'occasion **myC** vers du code C à 3 adresses. Le langage source proposé, un mini langage C, devra donc être compilé en C à 3 adresses.

Déroulé du projet et évaluation

Le projet de compilation se déroule du vendredi 8 novembre au lundi 25 novembre 2019, par groupe d'au plus deux élèves. Son évaluation s'appuiera sur:

- la qualité des échanges entre l'encadrant et les élèves pendant les séances de projet,
- la revue de code / micro-soutenance qui sera organisé avec chaque groupe lors de la dernière séance le 21 ou 22 novembre,
- la qualité des livrables déposés sur Thor au plus tard le lundi 25 novembre au matin.

Les groupes **devront être constitués** dès la fin de la première séance.

Ce projet a vocation à être réalisé de façon incrémentale, le traitement des spécifications de MyC étant intégrées petit à petit, comme le permet la description structurée de la syntaxe du langage.

Barème : le traitement correct du sous langage de MyC illustré par les points 1 à 4 ci-dessous (appelons le MyC-core) assurera une note d'au moins 11/20. Le traitement des extensions de MyC-core illustrés par les points 5 à 7, qui demande plus de réflexion et d'initiative, doit permettre d'aller au delà.

Les projets réalisés devront assurer:

- l'analyse des noms : les variables (et fonctions) utilisées sont-elles déclarées,
- l'analyse des types : les opérations effectuées sont-elles bien typées,
- la production de code à trois adresses.

Le langage MyC

MyC est un mini langage C qui vise à couvrir quelques éléments clés de la compilation. Il comprend notamment:

1. un mécanisme de déclarations explicite de variables,
2. des expressions arithmétiques arbitraires de type calculatrice,
3. des lectures ou écritures mémoires via des affectations avec variable utilisateur,
4. un mécanisme de typage comprenant notamment int et float,
5. des lectures ou écritures mémoires via des pointeurs,
6. définitions et appels de fonctions récursives,
7. un mécanisme de déclaration et d'utilisation de types structurés (struct),

Comme en C, la racine du programme à exécuter est décrite par la fonction *main* sans argument.

Le lexique du langage est précisé dans le fichier lang.l, la syntaxe du langage est précisée dans le fichier lang.y, tous deux disponibles sous [MyC.tar](#). Cette archive est complétée de quelques modules C dont les élèves sauront tirer profit.

Le code à trois adresses

Pour mémoire, le code à trois adresses se définit par des instructions C de la forme:

- $r = x$; (lecture mémoire)
- $x = r$; (écriture mémoire)
- $r1 = r2 \text{ op } r3$; (calcul)
- $r1 = *r2$; (lecture mémoire via pointeur)
- $*r1 = r2$; (écriture mémoire via pointeur)
- $r1 = (\text{type}) r2$; (cast explicite)
- $\text{goto } l$; (branchement inconditionnel)
- $\text{if } r \text{ goto } l$; et $\text{if } !r \text{ goto } l$; (branchement conditionnel)
- $t0 \text{ f } (t1 \ x1; t2 \ x2; \dots; tn \ xn) \{ \dots \}$; (définition de fonctions)
- $r = f(e1, e2, \dots, en)$; (appel de fonctions)

avec des variables utilisateurs x , des registres r , des opérations binaire op , des types t , des labels l etc, éventuellement augmenté de tout ce qui pourrait vous sembler nécessaire pour produire du code compilable par *gcc*.

On pourra inclure dans le code à trois adresses des *printf* permettant, à l'exécution, de vérifier la correction des calculs effectués.

Pour cela, à partir d'un code source **test.myc** on pourra produire 2 fichiers:

- un fichier **test.c** contenant les intructions 3 adresses produites, auxquelles seront ajoutés, aux endroits adéquats, les fonctions d'affichages (*printf*) et plus si besoin,
- un fichier **test.h** contenant toutes les déclarations nécessaires à la compilation de *test.c* tel que, par exemple, la déclarations des registres utilisés.

La compilation de **test.c** par *gcc* produira un exécutable **test** qui réalisera les calculs et les affichages éventuelles.

Livrables

Le code de chaque groupe est a mettre sur [thor](#) dans un répertoire contenant:

- un Makefile, pour compiler le compilateur en un exécutable *myc* se trouvant dans le repertoire courant,
- un script *compil* prenant en entrée le nom d'un fichier à compiler (ex. *test.myc*) et produisant, à l'aide de votre compilateur *myc* les fichiers compilés (*test.c* et *test.h*) et l'exécutable (*test*) issue de la compilation (*gcc*) de ces fichiers,
- un README, pour indiquer, en quelques lignes, l'étendu du compilateur réalisé,
- un sous-repertoire *src* qui contient les sources du compilateur,
- un sous-repertoire *test* qui contient un ensemble de fichiers sources (ex. *test.myc*) et les résultats de leurs compilations (*test.c*, *test.h* et *test*).

Les projets rendu doivent, a minima, compiler !