

Projet de compilation L3

Cahay Florian – Durand Florian

I. Compilation

Afin de compiler le compilateur de fichier tpc :

Il faut se rendre à la racine du dossier “ProjetCompilationL3_CAHAY_DURAND/” et exécuter la commande “make”.

II. Exécution d’un test unitaire

Afin d’exécuter un fichier de test précis :

Il faut se rendre à la racine du dossier “ProjetCompilationL3_CAHAY_DURAND/” et exécuter la commande “./bin/compil < chemin/NomDuFichier.tpc”.

Si aucune erreur n’est détectée, un fichier “prog.asm” va être créé dans le dossier “bin”.

III. Compilation du fichier “prog.asm”

Afin d’accéder aux fonctions “read_integer” et “print_integer” il est nécessaire de compiler “prog.asm” avec le fichier “utils.asm” situé dans le dossier “src”.

Ainsi un makefile a déjà été créé pour compiler “prog.asm”, celui-ci est également situé dans le dossier “src”.

Dès lors que le fichier “prog.asm” a été créé, il est possible de se rendre dans le dossier “src/” et exécuter la commande “make”.

Cela va créer un fichier exécutable dans le dossier “bin/”, il sera nommé “result.out”.

Il ne reste plus qu’à exécuter ce fichier pour vérifier le résultat produit par le fichier “prog.asm”.

IV. Exécution de tous les tests (compilation et exécution des fichiers prog.asm)

Afin d’exécuter tous les tests :

Il suffit de se rendre dans le dossier “tests/” et exécuter la commande “./test.sh” (certains fichiers peuvent demander d’entrer des valeurs, c’est pourquoi l’exécution se stoppe et attend qu’une valeur soit saisie).

A la suite de cela un fichier “log.txt” va être créé qui détaille les résultats du compilateur sur les fichiers.

V. Choix

- Le compilateur produit un arbre abstrait comme conseillé.
- Pour la table des symboles nous avons choisi d’utiliser une table de hachage avec des listes chaînées pour chaque valeur.

- Pour read et print un entier nous avons décidé de coder les fonctions en assembleur dans un fichier externe à celui produit par le compilateur.
- Nous avons modifié la grammaire en remplaçant respectivement *,/,% par STAR,SLASH,MOD afin de pouvoir récupérer la position précise du caractère.
- Nous avons utilisé une déclaration locale avec resb pour les variables globales.
- Les étiquettes sont nommées “L<numéro>” et les fonctions sont nommées “_nomDeLaFunction” dans le fichier assembleur.

VI. Difficultés

- Réussir à comprendre l’organisation de la mémoire et de la pile en assembleur.
- Comprendre les principes de table des symboles et arbre abstrait.
- Gérer les entiers avec 4 octets tout en conservant l’information du signe.
- Beaucoup d’imprévus qui font perdre du temps auxquels nous aurions dû penser en amont de la programmation.