

IFT209 –Programmation système

Devoir 4

Enseignants: Rosa Garcia

Date de remise: indiquée sur Turnin

Modalités de remise: Turnin

À réaliser: personne seule ou en équipe de deux

Thèmes : manipulation de chaînes de bits, récursivité, utilisation de la pile, interface avec le langage C.

Mise en situation

Un des derniers domaines où la connaissance des langages d'assemblage est cruciale est la fabrication de compilateurs. Bien que les compilateurs modernes soient fabriqués avec des outils flexibles permettant de passer facilement d'une architecture à une autre, il n'en demeure pas moins que le comportement et l'utilité de chaque instruction de l'architecture ciblée doivent être bien connus de ceux qui conçoivent le dit compilateur.

Plusieurs types d'énoncés peuvent se retrouver dans les langages évolués, certains servant à construire des structures de contrôle (if, while, for, switch), d'autres, déclarer des types donnés, écrire des expressions booléennes ou réaliser de simples affectations.

Compilation d'expressions arithmétiques

Un de ces types d'énoncés concerne l'arithmétique entière avec les opérateurs usuels (+,-,*,/), en notation infixée (l'écriture habituelle). Dans la majorité des langages évolués, on peut écrire des expressions telles que : $4+5-6$, ou bien, $(4-5)*7+9$, par exemple.

Transformer ces expressions en code machine n'est cependant pas une tâche simple. Tout d'abord, la notation infixée se prête mal à une génération directe d'instructions machine. La priorité des divers opérateurs doit être connue et prise en compte lors de l'évaluation de l'expression. Cependant, si l'expression est écrite dans la notation polonaise inverse, ce problème disparaît.

L'utilisation d'une structure de données pile est idéale pour évaluer le résultat, voici l'algorithme

1. Lire un par un les caractères de l'expression.
2. Si le caractère lu est un nombre alors on l'empile.

3. Si le caractère lu est un opérateur, alors on dépile les deux éléments se trouvant en haut de la pile et on calcule le résultat en appliquant l'opérateur sur les deux opérandes dépilés et on empile le résultat.
4. Une fois tous les caractères lus, la pile ne contient qu'un seul élément qui correspond au résultat final.

Notation infixée	Notation polonaise inverse
4+5-6	4 5 + 6 -

Avec une pile de données

	4	5	+	6	-
		5		6	
	4	4	9	9	3

Notation infixée	Notation polonaise inverse
(4-5)*7+9	4 5 - 7 * 9 +

Avec une pile de données

	4	5	-	7	*	9	+
		5		7		9	
	4	4	-1	-1	-7	-7	2

Tâche à réaliser

Pour réussir ce devoir, vous devez réaliser en assembleur de ARM v8 l'évaluation des expressions en notation polonaise inverse avec une pile virtuelle.

Votre programme évalue uniquement des expressions arithmétiques contenant des constantes positives et les quatre opérateurs +, -, *, et /.

Votre exécutable final doit répondre aux critères suivants :

- 1- Lire une chaîne de caractères représentant une expression arithmétique en notation polonaise inverse. L'expression peut contenir uniquement les opérateurs +, -, *, et /, ainsi que des **entiers positifs**.
- 2- Évaluer le résultat en utilisant une pile
- 3- Afficher le résultat

Quelques conseils pratiques

Sauvegardez sur la pile uniquement les registres qui sont utilisés!
Rappelez-vous bien l'ordre dans lequel vous devez empiler et dépiler!

Bon travail!