

Vendredi 12 Janvier 2007

Projet de Traduction

Daniel BORGES
Sébastien PLANSON

SOMMAIRE

I. Utilisation

II. Notes de compilation

III. Bugs connus

I. Utilisation

Le programme `tcompil` prend en paramètre un fichier dans le langage TPC ou le lit depuis l'entrée standard si aucun fichier n'est spécifié et le compile dans le langage de la machine virtuelle fournie. Le résultat est imprimé sur la sortie standard ou dans un fichier si l'option `-o` est spécifiée.

Exemple :

```
$ cat test.tpc
main()
{
    write(5);
}
$ ./tcompil test.tpc
      0      JUMP      2      #goto 0
#label: 0
#--- main ---
#--- set ---
      2      SET      5
#--- write ---
      4      WRITE
      5      HALT
$ ./tcompil test.tpc -o
$ ../VM/vm test.vm

Write:5
```

Quelques exemples de programmes sont fournis dans le répertoire `exemples`.

II. Notes de compilation

Le programme « assembleur » créé commence toujours par un JUMP vers la première instruction du main.

Les valeurs de retour des fonctions sont toujours placées sur la pile. Pendant l'exécution d'une fonction, la valeur de retour est donc toujours placée à l'adresse -3.

Les arguments d'une fonction sont empilés avant l'appel d'une fonction. Ils sont donc accessibles pendant l'exécution d'une fonction à partir de -3, sauf si la fonction retourne une variable auquel cas ils sont accessibles à partir de -4. Les arguments restent dans l'ordre du source TPC.

Exemple :

```
int pgcd(n,d)
```

```
...
```

La valeur de retour sera en -3.

n sera en -5.

d en -4.

En cas d'erreur lors de la compilation, les différentes tables des symboles sont affichées.

Les tables des symboles sont affichées de la manière suivante :

```
[<adresse>] <nom> : <signature> (size : <taille>)
```

<adresse> représente l'adresse dans la pile du début d'une variable ou la ligne à laquelle débute une fonction.

<nom> est le nom du symbole.

<signature> est le nombre d'arguments d'une fonction si le symbole est une fonction, -1 s'il s'agit d'une variable et la valeur de la constante s'il s'agit d'une constante.

<taille> est la taille du symbole :

- 1 pour une variable,
- n pour un tableau de taille n,
- 0 pour une constante,
- 1 pour une fonction qui retourne une valeur et 0 sinon.

Sont d'abord donnés les variables, fonctions et constantes globales, puis les variables et constantes locales à la fonction en cours de compilation.

III.Bugs connus

Les affectations dans des tableaux semblent bugguées mais il est pourtant tout à fait possible de faire un `read()` sur un tableau.

`T[3] = 5;` donnera un résultat incohérent à l'exécution mais `read(T[3]);` fonctionnera parfaitement...

Les exceptions du langage TPC ne sont pas gérées.

Les fonctions prenant en paramètre des tableaux ne compilent pas correctement.

Une variable ne peut être déclarée après un tableau...