Architecture des ordinateurs

TP 1 : Assembleur Sparc

Arnaud Giersch, Benoît Meister et Frédéric Vivien

Remarques préliminaires :
- le nom d'un fichier contenant un programme assembleur doit obligatoirement avoir le suffixe `.s` ;
- on passe d'un programme assembleur à un fichier exécutable en utilisant `gcc` de la même manière qu'avec un programme C.

1. Étude d'un programme en assembleur Sparc

   (a) Récupérez le programme `addition.s` qui réalise la somme de 2 entiers. Compilez en utilisant la commande `gcc -o addition addition.s` et exécutez-le.

   URL : `http://icps.u-strasbg.fr/~vivien/Enseignement/Archi-2001-2002/addition.s`.

   Étudiez ce programme et rajoutez des commentaires explicatifs dans ce fichier addition.s.

   (b) Réalisez un programme C faisant appel à une fonction prenant 8 paramètres, produisez le programme assembleur correspondant (option `-S` de `gcc` : `gcc -S fichiersource`).

   Déterminez ensuite quels paramètres sont placés dans les registres, lesquels ne le sont pas et trouvez l'emplacement mémoire de ces derniers.

2. Exercices de programmation

   (a) Écrivez un programme assembleur calculant la factorielle d'un entier de manière itérative (une seule fonction principale contenant une boucle).

   (b) Écrivez un programme assembleur calculant la factorielle d'un entier de manière *récursive*.

   (c) Modifiez-le programme précédent pour qu'il affiche à chaque étape de la récursion les valeurs des pointeurs de pile (`%sp` et `%fp`).

   (d) Dans un processeur où la multiplication n'est pas implémentée par un circuit, celle-ci peut être réalisée efficacement en se basant sur l'algorithme suivant :

$$
\begin{aligned}
a \times b := \;& \textbf{si } b = 0 \textbf{ alors } 0 \\
& \textbf{sinon si } b = 2 \times b' \textbf{ alors } (a \times 2) \times b' \\
& \textbf{sinon } ((a \times 2) \times b') + a
\end{aligned}
$$

   En vous appuyant sur cette méthode, proposez une fonction assembleur réalisant la multiplication entière en utilisant uniquement des additions et des décalages.

   (e) Utilisez la fonction précédente dans un programme C. L'exécutable s'obtient en donnant simplement le fichier C et le fichier assembleur au programme `gcc`, ce dernier s'occupe de faire les liens.

   (f) Écrivez un programme assembleur de recherche de l'élément minimum d'un tableau. Le tableau est une variable locale à la routine principale. Cette dernière fait appel à une routine pour la recherche de l'élément minimum d'un tableau.