université angers
FACULTÉ DES SCIENCES
Unité de formation et de recherche

Département Informatique
L3 Mention Informatique
Jean-Michel Richer
Architecture des Ordinateurs
2010/2011

jean-michel.richer@univ-angers.fr

Projet de Programmation

1 But du travail

Le but de ce travail est de vous familiariser avec la programmation assembleur et en particulier avec les instructions des unités SSE.

2 Objectif

Réaliser en assembleur Pentium une calculatrice vectorielle pour les nombres flottants en simple précision qui fonctionne suivant le principe d'une pile :

- tout nouveau vecteur est placé au sommet de la pile
- toute opération utilise les données placées dans la pile et commence par traiter les opérandes au sommet de la pile
- le résultat d'une opération est placé au sommet de la pile

Les données en entrée sont des vecteurs de 4 nombres flottants en simple précision qui tiennent donc dans un registre SSE.

Les opérations à implanter sont les suivantes :

Description	Symbole	Comportement
addition	$+$	$\forall i \in [0,3],\ z_i = x_i + y_i$
soustraction	$-$	$z_i = x_i - y_i$
multiplication	$*$	$z_i = x_i \times y_i$
division	$/$	$z_i = x_i / y_i$
racine carrée	$SQRT$	$z_i = sqrt(x_i)$
norme	$NORME$	$z_i = sqrt(\sum_{i=0}^{i=3} {x_i}^2)$

La calculatrice dispose en outre de 16 registres mémoire de 128 bits qui permettent de stocker des données de manière temporaire. On dispose de 2 instructions de manipulation de ces registres :

- `STORE` n : supprime la donnée en sommet de pile et la stocke dans le registre n ($n = 0, \ldots, 15$)
- `LOAD` n : mettre la donnée du registre n au sommet de la pile

ainsi que d'instructions pour l'affichage

- `PRINT` n : afficher la donnée stockée dans le registre n
- `PRINT` : affiche la donnée en sommet de pile

3 Données en entrée

L'exécutable devra s'appeler `calcvect` et prendra comme argument un nom de fichier contenant des opérations et données, comme suit :

```
; le point virgule indique un début de commentaire
; définition d'une donnée (vecteur de 4 float), placé en sommet de pile
[12.0, 13.1, 0, 1.0]
; on dépile et on stocke dans le registre 0
STORE 0
[ 3.0,  2.2, 0, 0]
STORE 1
[2, 3, 1, 2]
STORE 2
LOAD 0
LOAD 1
-
PRINT
LOAD 2
/
PRINT
NORME
PRINT
```

L'affichage obtenu à l'écran devrait être le suivant :

```
[9.0, 10.9, 0, 1.0]
[4.5, 5.45, 0, 0.5]
[7,085372, 7,085372, 7,085372, 7,085372]
```

Le programme commencera par détecter la présence des instructions SSE grâce à l'instruction CPUID. Dans le cas ou les instructions SSE ne sont pas présentes, on sortira du programme en affichant un message d'erreur.

4 Contraintes

- Ecrire le programme pour l'assembleur NASM sous Linux
- Vous pouvez réaliser l'édition de liens avec gcc et la bibliothéque C. Les seules commandes externes que vous pouvez utiliser sont : **printf, scanf, malloc, free, fopen, fclose, fscanf**. Toutes les autres opérations doivent êtes écrites en assembleur.
- Le programme final est à rendre pour le 25 Décembre 2010, 11h59 dernier délai. Au dela de cette date, vous obtenez la note de 0. La note que vous obtiendrez sera prise en compte dans la note de contrôle continu.
- vous pouvez étendre le langage initial, mais pas le modifier
- Vous pouvez travailler par groupe de 2 étudiants maximum

Vous pourrez rendre votre travail en m'envoyant un fichier archive compressé (commande tar) de vos fichiers sources (l'option `z` indique que le fichier archive est compressé) :

```
tar -cvzf calcvect.tgz *
```

à l'adresse suivante en indiquant **les noms des étudiants** ayant travaillé sur le projet :

`jean-michel.richer@univ-angers.fr`

En retour, je vous enverrai mes commentaires et remarques quant à votre travail en vous indiquant ce qui est bien et ce qui l'est moins.

Que la force soit avec vous jeunes padawans !