

Travaux Pratiques d'Architecture des ordinateurs

Licence 2
Faculté des Sciences et Technologie, UPEC

2015

Les TP s'effectuent sous Linux. L'assembleur utilisé est `yasm`.

Premier programme

1. Télécharger les fichiers `first.asm`, `driver.c`, et `Makefile`.

Les commandes de compilation sont :

```
gcc -c -g -std=c99 -m64 driver.c
yasm -g dwarf2 -f elf64 first.asm
gcc -m64 -g -std=c99 -o first driver.o first.o
```

On peut compiler tous les exemples avec la commande `make`.

2. Exécuter le programme `first`.
3. Exécuter `gdb first`, lancer les commandes `break 1` puis `run`. Exécuter ensuite le programme pas à pas en utilisant `step`.
On peut afficher le contenu d'un registre avec `print $nomDuRegistre` et le contenu d'une case mémoire [*adresse*] avec `print adresse`.
Si un tableau est stocké à l'adresse *adresse*, `print adresse@nombreElement` affiche *nombreElement* du tableau, `print (&adresse+numeroCase)` affiche le contenu de la case *numeroCase* du tableau.

Exercices (4 séances de TP)

1. Écrire un programme de saisie de 2 entiers *a* et *b*, et d'affichage de *a* OR *b*, de *a* AND *b*, de *a* XOR *b*.
2. Écrire un programme qui demande à l'utilisateur si c'est un homme ou une femme, lit sa réponse (0 pour un homme, 1 pour une femme) et affiche selon le cas "Bonjour monsieur" ou "Bonjour madame".
3. Écrire un programme qui prend en entrée un entier signé *n* et affiche s'il est pair ou impair en utilisant l'instruction `and` et un masque.
4. Reprendre la question précédente en utilisant l'instruction `idiv`.
5. Écrire un programme demandant un entier à l'utilisateur et testant si cet entier est une puissance de 2.
6. Écrire un programme calculant le produit de 2 entiers positifs en utilisant une boucle, des décalages et des additions (le produit est supposé tenir sur 64 bits). Il devra permettre la saisie des 2 opérandes, et finir par l'affichage du résultat.
7. Modifier le programme de l'exercice précédent pour qu'il puisse aussi prendre en compte des opérandes négatifs.
8. Écrire un programme demandant un entier à l'utilisateur et testant si son écriture binaire est de la forme : $0^k 1^n 0^n$, c'est à dire si elle se compose (de gauche à droite) d'un certain nombre *k* de zéros, puis de *n* uns suivis de *n* zéros.
9. Transformer le programme de l'exercice 7 pour que le calcul soit réalisé par une fonction. La saisie des deux opérandes, l'appel de la fonction produit et l'affichage du résultat se feront dans le programme principal. Vérifier que l'on peut appeler la fonction plusieurs fois dans le programme principal.

10. Écrire un programme qui demande une chaîne de caractères à l'utilisateur, vérifie si la chaîne représente un nombre hexadécimal, calcule la valeur du nombre et affiche le résultat contenu dans un registre en base 10 et en hexa.
11. Écrire un programme qui prend en entrée un entier signé et qui dit si sa représentation binaire correspond à une expression bien parenthésée, le 0 correspondant à une parenthèse ouvrante et le 1 à une parenthèse fermante.
12. Écrire un programme avec une fonction qui prend en paramètre un tableau d'entiers et le nombre d'éléments du tableau et renverse l'ordre des éléments dans le tableau. Tester la fonction sur différents tableaux dans le programme principal.
13. Écrire un programme qui calcule le triangle de Pascal jusqu'à $\binom{15}{15}$ c'est à dire qui calcule $\binom{n}{p}$ pour n et p allant de 1 à 15 et stocke les valeurs dans un tableau. Le programme demande ensuite des valeurs de n et p , cherche la valeur de $\binom{n}{p}$ dans le tableau et affiche le résultat.