
Travaux dirigés 1 (double) : Premier programme C - affectation

1 Un premier programme C

```
1  /* Declaration de fonctionnalités supplémentaires */
2  #include <stdlib.h> /* EXIT_SUCCESS */
3
4  /* Declaration des constantes et types utilisateurs */
5
6  /* Declaration des fonctions utilisateurs */
7
8  /* Fonction principale */
9  int main()
10 {
11     /* Declaration et initialisation des variables */
12     int x = 5;
13     int y;
14
15     y = 2;
16     x = y;
17
18     /* valeur fonction */
19     return EXIT_SUCCESS;
20 }
21
22 /* Définitions des fonctions utilisateurs */
```

FIGURE 1 – Un programme C et sa traduction machine

Voici, figure 1, un premier programme C. Le langage C sera présenté au prochain cours. Pour le moment voici ce que vous avez à savoir.

Un programme en langage C est donné sous forme de texte, *le code source*, et il doit passer par une étape de traduction avant de pouvoir être exécuté par le processeur. Par analogie avec la musique, le code source est la partition, et le fichier exécutable est le morceau musical (codé sur le support adapté au système de lecture : un fichier mp3, un CD, etc.). La traduction est effectuée par un ensemble de programmes (et non par des musiciens), le code source doit donc obéir à des règles syntaxiques précises.

Les textes entre */** et **/* sont des *commentaires*, ils ne feront pas partie du programme exécutable, ils servent aux humains qui manipulent les programmes. Les commentaires du programme (figure 1) vous serviront au cours du semestre à structurer tous vos programmes C.

De même, le programme est indenté : le développeur a choisi d'ajouter quelques caractères espaces (ou des tabulations) en début de certaines lignes d'instruction (lignes 11 à 19). L'objectif de l'indentation est d'aider la lecture du code en faisant ressortir des blocs de code. En langage C, l'indentation n'a aucun impact sur la syntaxe et le programme exécutable.

Tout programme C comporte une fonction principale, le `main()`, qui sert de point d'entrée au programme. Cette fonction doit se terminer par l'instruction `return EXIT_SUCCESS`. En renvoyant cette valeur, le `main` signale au système d'exploitation la terminaison correcte du programme.

Une *variable* est un symbole (habituellement un nom) qui renvoie à une position de mémoire dont le contenu peut prendre successivement différentes valeurs pendant l'exécution d'un programme¹.

L'instruction `int x = 5` déclare une variable `x` et fixe sa valeur initiale à 5. Le mot clé `int` signifie que cette variable contiendra un entier.

L'instruction `int y` déclare une variable entière `y` sans l'initialiser. L'effet de cette déclaration est de réserver un espace mémoire pour `y` stocker un entier.

Le signe égal (=) a un sens bien particulier, il dénote une *affectation*. L'objectif de ce TD est de bien comprendre l'affectation. La partie à gauche du signe égal doit désigner un espace mémoire, c'est typiquement une variable. La partie à droite du signe égal est une expression dont la valeur sera évaluée et écrite à l'adresse à laquelle renvoie la partie gauche. Par exemple, `y = 2` a été traduit en code machine par une instruction évaluant l'expression 2 dans un registre, et par une instruction d'écriture de la valeur trouvée dans la mémoire réservée à `y`. Une variable s'évalue comme sa valeur (celle contenue dans la mémoire correspondante, au moment de l'évaluation).

Questions

1. Expliquer pas à pas chaque ligne de ce programme et écrire sa trace.
2. À la ligne 15 dans le programme C, on remplace l'instruction `y = 2` par `y = x + 2` (on considère qu'on dispose de plusieurs registres). Quelle est la trace en C ?

1. <http://fr.wikipedia.org/wiki/Variable>

2 Échange de valeurs

2.1 Introduction au problème

Nous avons deux tableaux anciens, chacun accroché à un clou, et un troisième clou, libre, sur le mur d'une exposition.

Pour des critères esthétiques, nous voulons changer de place nos deux tableaux sans les mettre par terre, car cela risquerait d'abîmer nos précieuses toiles... Comment faire ?



FIGURE 2 – Échanger les deux tableaux en utilisant le clou libre...

2.2 Échange des valeurs de deux variables en C

1. En utilisant la solution de l'exercice précédent, proposer un algorithme qui déclare et initialise deux variables entières, puis qui effectue la permutation de ces deux valeurs.
2. En utilisant l'algorithme défini précédemment, écrire un programme C effectue la permutation des deux valeurs contenues dans les variables x et y .
3. Exécuter le programmes sur un exemple (faire une trace).

3 Indentation d'un programme C

Indenter correctement le programme suivant :

```
#include <stdlib.h>  int main (  
) {int x; int y; int z; x=10;  
y = 30; z = x + y; y = y / 3;  
x = x * x; return  
EXIT_SUCCESS;}
```