

# Introduction à la programmation

Jean-Sébastien Coron

Université du Luxembourg

# Introduction à la programmation

- ▶ Enseignement :
  - ▶ Cours: Jean-Sébastien Coron et Benoit Cogliati
  - ▶ TD: Moon Sung Lee
- ▶ But du cours :
  - ▶ Apprendre les bases de la programmation d'un ordinateur.
  - ▶ Se familiariser avec le langage C.

# Plan

- ▶ Introduction:
  - ▶ Qu'est que l'informatique ?
  - ▶ Qu'est qu'un ordinateur ?
- ▶ Les langages de programmation
- ▶ Application: le langage C.
  - ▶ Présentation du langage.
  - ▶ Le typage des données. Les opérateurs.
  - ▶ L'affichage.

# Qu'est-ce que l'informatique

- ▶ Informatique = traitement automatisé de l'information par ordinateur.
- ▶ L'information: l'ordinateur manipule des nombres qui représentent différent type d'information:
  - ▶ Nombres (calcul scientifique)
  - ▶ Lettres (traitement de texte)
  - ▶ Image, son.
- ▶ Automatisé: le traitement est effectué par un programme qui se déroule tout seul.
  - ▶ L'utilisateur donne des paramètres en entrée et reçoit un résultat en sortie.

# Traitement des données

- ▶ Création de données
  - ▶ Nombres: acquisition (par exemple la vitesse sur un avion)
  - ▶ Texte: l'utilisateur tape un texte.
- ▶ Analyse des données:
  - ▶ Nombres: calculs sur ces nombres (par exemple calcul de la puissance nécessaire).
  - ▶ Texte: occurrence d'un mot dans un traitement de texte.
- ▶ Modification des données:
  - ▶ Nombres: calcul
  - ▶ Texte: modification d'un texte.

# Qu'est-ce qu'un ordinateur

- ▶ Ordinateur: ensemble de circuits électroniques manipulant des données binaires.
- ▶ Processeur:
  - ▶ Effectue les opérations de base.
  - ▶ Ordinateur moderne:  $10^9$  opérations par seconde.
- ▶ Mémoire :
  - ▶ Stockage des données dont l'ordinateur a besoin.
- ▶ Entrées-sortie.
  - ▶ Communiquer avec l'extérieur.
  - ▶ Entrées: clavier, souris.
  - ▶ Sortie: écran.

# Langage de programmation

- ▶ L'ordinateur ne comprend que le langage machine.
  - ▶ Opérations élémentaires sur des données binaires.
  - ▶ Spécifiques à chaque processeur.
  - ▶ Difficile et pénible à utiliser.
- ▶ Langage de programmation.
  - ▶ Facilite la programmation.
  - ▶ Interface homme-machine.

# Programmation impérative

- ▶ Forme de programmation la plus répandue.
  - ▶ Exemple: Assembleur, BASIC, C, C++, Pascal.
- ▶ Décrit la programmation comme une suite d'instructions exécutés par l'ordinateur pour traiter des données.
- ▶ Types d'instruction :
  - ▶ Instructions d'assignation: stockage d'une donnée en mémoire.
  - ▶ Branchement conditionnel: exécution d'une suite d'instruction si une certaine condition est réalisée.

# Langage compilé, interprété

- ▶ Langage interprété
  - ▶ Un interpréteur est un programme qui lit et exécute au fur et à mesure les instructions du langage
  - ▶ Exemple: maple.
- ▶ Langage compilé:
  - ▶ Un compilateur traduit la séquence d'instructions de haut niveau en des instructions du microprocesseur.
  - ▶ On exécute ensuite le programme résultat de la compilation.
  - ▶ Exemple: langage C.

# Le langage C

- ▶ Langage de programmation le plus populaire.
- ▶ Programmation structurée.
  - ▶ Programmation par procédures.
- ▶ Typage de données.
  - ▶ Types proches de la machine (int, float...).

# Programme "Hello world"

```
#include <stdio.h>
int main()
{
    printf("Hello world \n");
}
```

- ▶ `#include`: inclusion d'autres modules ou librairies.
  - ▶ `stdio.h`: librairie d'entrées/sorties.
- ▶ `int main()`: définition de la fonction principale.
- ▶ `printf()`: instruction d'affichage.
  - ▶ `\n`: retour à la ligne.

# Utilisation

- ▶ Saisir le programme.
  - ▶ `xemacs hello.c &`.
- ▶ Le compiler.
  - ▶ `gcc hello.c -o hello.exe`
- ▶ L'exécuter.
  - ▶ `./hello.exe`

# La fonction main

```
#include <stdio.h>
int main()
{
    printf("Hello world \n");
}
```

- ▶ Fonction main:
  - ▶ C'est la fonction appelée lors du lancement du programme.
  - ▶ Aucune autre n'est exécutée automatiquement.
- ▶ Indispensable.

# Variables

- ▶ Le programme peut stocker des variables en mémoire.

```
int a;  
a=2;  
printf("a=%d\n",a);
```

- ▶ Déclaration d'un entier: `int a;`
- ▶ Assignment: `a=2;`
- ▶ Affichage:
  - ▶ `printf` avec `%d` pour un entier.
  - ▶ Résultat: `a=2`

# Typage

- ▶ Il faut définir le type des variables.
- ▶ Variables entières.
  - ▶ short: 16 bits  $\pm 32767$ .
  - ▶ int: 16 ou 32 bits  $\pm 2 \cdot 10^9$ .
  - ▶ long: 32 bits  $\pm 9 \cdot 10^{18}$ .
- ▶ unsigned short, unsigned int, unsigned long → entiers positifs.

# Les flottants

- ▶ Codage:
  - ▶ Mantisse:  $m$ .
  - ▶ Exposant:  $e$ .
  - ▶  $m * 2^e$ .
- ▶ float: 24+8 bits.
  - ▶ Précision  $2^{-23}$ ,  $< 10^{38}$ .
- ▶ double: 53+11 bits.
  - ▶ Précision  $2^{-53}$ ,  $< 10^{308}$ .
- ▶ long double: 64+16 bits.
  - ▶ Précision  $2^{-64}$ ,  $< 10^{4932}$ .

# Déclaration des variables

- ▶ Variables globales : au début du programme, valables dans tout le programme.
- ▶ Variables locales : à l'intérieur d'une fonction, valables à l'intérieur de la fonction.

```
#include <stdio.h>
int u;
int main()
{
    int a;
    a=2;
    u=3;
    printf("a=%d,u=%d\n",a,u);
}
```

# Initialisation des variables

- ▶ Lors de la déclaration d'une variable, son contenu est aléatoire.
- ▶ On peut l'initialiser simultanément.

```
#include <stdio.h>
int u=3;
int main()
{
    int a=2;
    printf("a=%d,u=%d\n",a,u);
}
```

# Opérateurs sur les nombres

- ▶ On peut réaliser les opérations suivantes sur les entiers et les flottants.
  - ▶  $a + b$ : addition.
  - ▶  $a - b$ : soustraction.
  - ▶  $a * b$ : multiplication.
  - ▶  $a / b$ : division.
    - ▶ Division euclidienne pour les entiers.
  - ▶  $a \% b$ : reste modulaire.

## Affichage des variables

- ▶ L'instruction `printf` affiche sur la sortie standard (l'écran) du texte et le contenu des variables.
  - ▶ `%d` pour un `int` ou `long`.
  - ▶ `%f` pour un `float` ou `double`.

```
float a=2.3;  
int b=4;  
printf("a=%f,b=%d\n",a,b);
```