Séance 1

Introduction et bases de la programmation, partie 1

Sébastien Combéfis

mercredi 10 septembre 2014

Contenu

Concepts de base en programmation, appliqué avec le langage C

- Bases de la programmation : variables, conditions et boucles
- Représentation des données en binaire
- Tableaux à une et à deux dimensions
- Procédures et fonctions
- Découpe en sous-problèmes et librairie standard
- Structure de la mémoire et pointeurs
- Structures et listes chainées
- Manipuler des fichiers

Activités

- Dix-huit séances de cours
  - Douze séances théoriques avec exercices en salle
  - Quatre séances d'exercices sur machine
  - Une séance de révision
- L'examen se déroule la dernière séance du 28 janvier 2015
  - Examen sur papier, durée 2h30
- Régulièrement des petits tests et devoirs

Évaluation

- **Première session**
  - L'examen compte pour 60% de la note finale
  - Les tests et devoirs comptent pour 40% de la note finale
- **Deuxième session**
  - L'examen compte pour 100% de la note finale
  - Sauf si compter les tests et devoirs augmente la moyenne

Ressources

- Documents relatifs au cours disponibles sur :

  http://sebastien.combefis.be/teaching/itscm/prog

- Syllabus provisoire
- Livres de référence

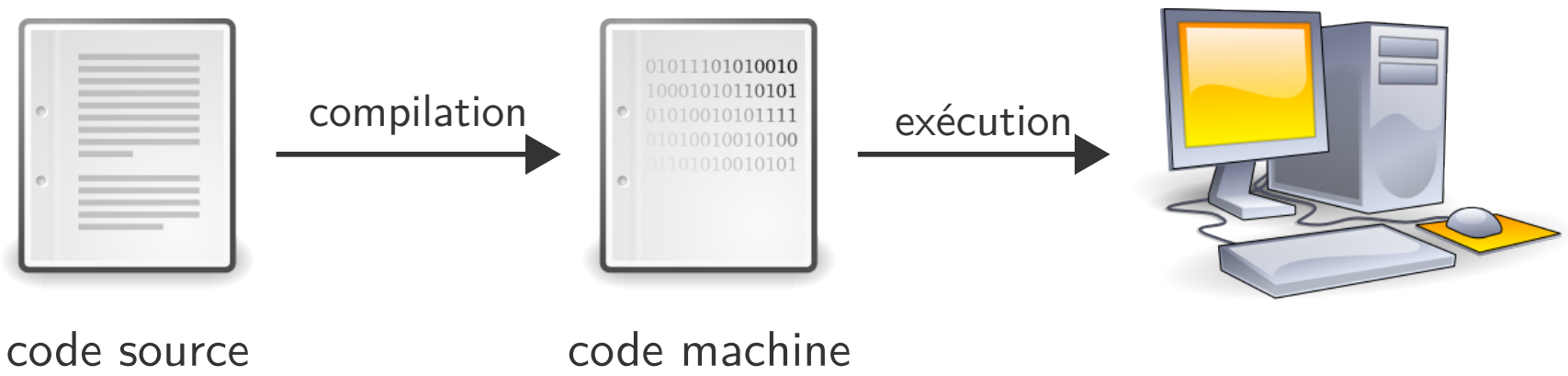
Programme

- Un programme est une séquence d'instructions
  - Il reçoit des données en entrée
  - Il effectue des calculs
  - Il produit un résultat en sortie
- Plusieurs manières de fournir les entrées, produire les sorties

  Paramètre ligne de commande, demande à l'utilisateur, fichier...

Chaine de compilation

- Le code source est écrit dans un langage de programmation
- Le compilateur transforme le code source en code machine
- Le code machine est exécuté par l'ordinateur



Hello World

```c
#include <stdio.h>

int main()
{
    printf ("Hello World\n");    // Affiche "Hello World"

    return 0;
}
```

- `#include` inclus du code existant qu'on peut utiliser

  Afin de pouvoir utiliser `printf()`

Hello World

```c
#include <stdio.h>

int main()
{
    printf ("Hello World\n");    // Affiche "Hello World"

    return 0;
}
```

- `int main()` est une fonction qui retourne une valeur

  Le contenu de cette fonction est le programme même

Hello World

```c
#include <stdio.h>

int main()
{
    printf ("Hello World\n");    // Affiche "Hello World"

    return 0;
}
```

- `printf("")` permet d'afficher du texte à l'écran

  \n permet d'effectuer un retour à la ligne

Hello World

```c
#include <stdio.h>

int main()
{
    printf ("Hello World\n");    // Affiche "Hello World"

    return 0;
}
```

- // permet d'écrire un commentaire

  N'influence pas le programme

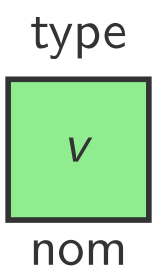
Hello World

```
1  #include <stdio.h>
2
3  int main()
4  {
5      printf ("Hello World\n");    // Affiche "Hello World"
6
7      return 0;
8  }
```

- `return 0` doit être fait à la fin de la fonction `main()`

Variable

- Un programme manipule des données
- Les données sont stockées dans des variables
- Une variable possède :
  - Un nom
  - Un type
  - Une valeur



Déclaration de variables

- Avant de pouvoir utiliser une variable, il faut la déclarer
- Une variable peut être non-initialisée
- La valeur d'une variable peut être modifiée à tout moment

```
1  int a = 1;
2
3  int b;
4  b = 3;
5
6  a = 5;
```

La fonction `printf` I

- Permet d'afficher du texte à l'écran
- `printf("%d", var)` affiche la valeur d'une variable

  %d sera remplacé par la valeur de la variable `var` de type `int`

```
1  printf ("Hello");              // Affiche "Hello"
2
3  int age = 20;
4  printf ("Âge %d", age);        // Affiche "Âge 20"
```

La fonction printf II

- On peut aussi afficher la valeur de plusieurs variables

```
1  int annee = 1978;
2  int mois = 7;
3  int jour = 11;
4
5  printf ("Je suis né le %d/%d/%d", jour, mois, annee);
6
7  // Affiche "Je suis né le 11/7/1978"
```

Opérations de base

- Addition, soustraction et multiplication : +, - et *
- Division : /

  Calcule la division entière

- Modulo : %

  Calcule le reste de la division entière

```
1  int a;
2  a = 2 + 3;                    // a vaut 5
3
4  int b = 4 / 2;                // b vaut 2
5
6  int c = a % b;                // 5 % 2, le reste vaut 1
```

Les conditions

- a strictement plus grand/strictement plus petit que b :

  `a > b` `a < b`

- a plus grand ou égal/plus petit ou égal que b :

  `a >= b` `a <= b`

- a égal à/différent de b :

  `a == b` `a != b`

```
1  int a = (5 >= 2);          // a vaudra 1
2
3  int b = (5 <= 1);          // b vaudra 0
4
5  int c = (a == b);          // c vaudra 0
```

Instruction `if`

- Exécute du code en fonction d'une condition

  Si la valeur est nulle, le code n'est pas exécuté

  Pour les autres valeurs, le code est exécuté

```
1  int distance = 10;
2
3  if (distance > 5)
4  {
5      printf ("La distance est plus grande que 5");
6  }
7
8  printf ("La suite...");
```

Instruction if-else

- Définir un code si la condition est vérifiée et un autre sinon

```
1  int distance = 10;
2
3  if (distance > 5)
4  {
5      printf ("La distance est plus grande que 5");
6  }
7  else
8  {
9      printf ("La distance est plus petite que 5");
10 }
11
12 printf ("La suite...");
```

Boucle `while`

- Répète des instructions tant qu'une condition est satisfaite
- Forme générale :

  `while (`*`condition`*`)`

```
1  int i = 3;
2
3  while (i > 0)
4  {
5      printf ("%d", i);
6      i = i - 1;
7  }
8
9  printf ("BOUM !");
```

Boucle `for`

- Répète des instructions un certain nombre de fois
- Forme générale :

  `for (`*initialisation*`;` *condition*`;` *mise à jour*`)`

```
1  int i;
2
3  for (i = 3; i > 0; i = i - 1)
4  {
5      printf ("%d", i);
6  }
7
8  printf ("BOUM !");
```

Exécution d'une boucle

- Les boucles `while` et `for` sont similaires

```
for (init; cond; upd)
{
    body;
}
```

```
init;
while (cond)
{
    body;
    upd;
}
```