Paradigme de programmation Impérative

DR. Sofiane AOUAG
Université De Batna 2
Faculté M-I– Département Informatique

2020/2021

Plan du cours

- Introdcution
- Variable et assignement
- Système de type
- Structures de contrôle
- Les échappement (Escape)
- Difficultés liés au paradigme impératif
  - Les effets de bord
  - Synonymie (Aliasing)
  - Exception
- Conclusion

Introduction

- Programme = Séquence d'actions (Ordre): D'où le terme IMPERATIF
- La programmation impérative est caractérisée par, la programmation avec un état et des commandes qui modifient l'état.
- L'état d'exécution d'un programme est matérialisé par :
  - Un point de contrôle (où se trouve l'exécution);
  - Contenu de toutes les variables;
  - Entrée qui reste à lire et sortie qui reste à produire.
- Après chaque opération, l'état change. Un changement d'état est déterminé par des opérations d'assignements et une séquence de commandes
- L'ordre des opérations est important: L'exécution est généralement séquentielle: instruction, instruction qui suit, …

Introduction

- L'ensemble de commandes permettent de structurer le code et de manipuler la mémoire.
- La programmation impérative est basée sur le modèle machine de Von Neumann.
- Dans une machine Von Neumann, on distingue traditionnellement deux parties:
  - L'unit centrale qui assure le stockage et le traitement des données ainsi que le stockage des programmes et des résultats.
  - L'unité d'échange qui a en charge des communications avec l'extérieur : entrée des données et des programmes; sortie des résultats. Le clavier, le moniteur, la souris et l'ensemble des périphériques relèvent donc de cette notion abstraite d'unit d'échange.

Variable et assignement

- Un programme est conceptualisé comme une suite d'instructions pour une machine de Von Neumann

Exemple :

Lire X, Lire Y;

Calculer X*Y;

Mettre le résultat dans Z:=X*Y;

  - Variable = cellule mémoire : Référence à une variable = charger (LOAD)
  - Affectation d'une variable = enregistrer (STORE)
  - Structures de contrôle = sauts (JUMPS)
- Une instruction du langage correspond à un ensemble d'instructions du langage machine

Variable et assignement

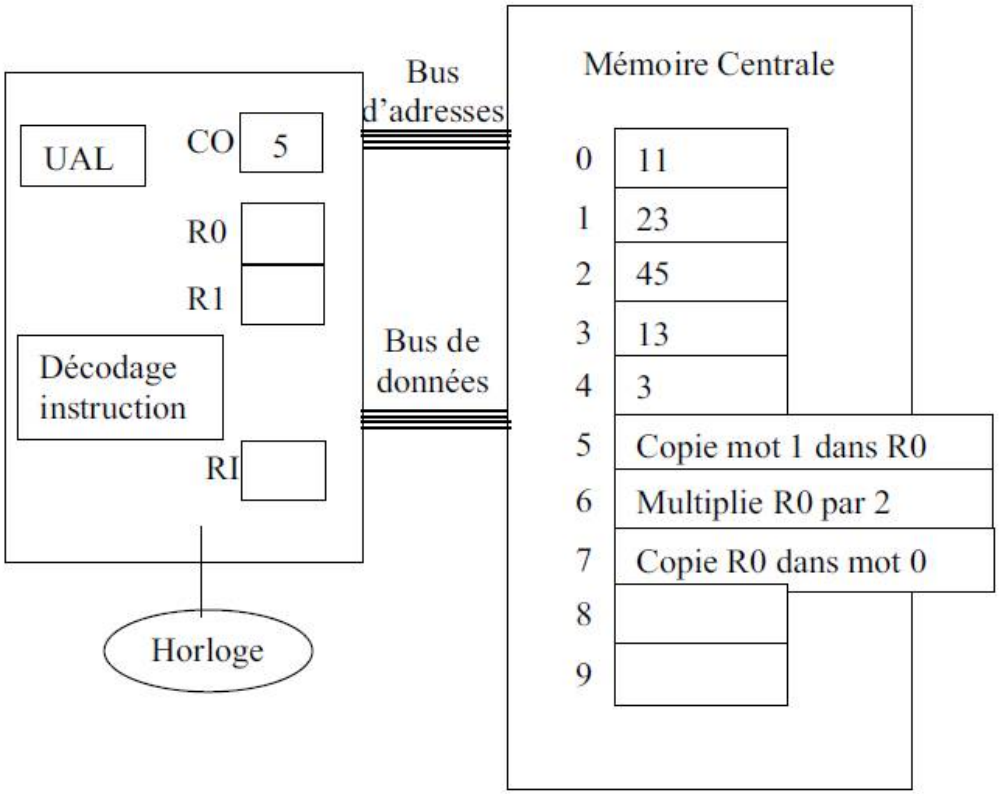

Schéma d'une unité centrale + mémoire centrale contenant un programme.

Variable et assignement

- La plupart des descriptions des Langages de programmation impératifs sont liées au matériel et aux considérations d'implémentation où:
  - Un nom est lié à une adresse,
  - Une variable à une cellule mémoire,
  - Et une valeur à une configuration binaire

- Une variable peut être liée à un emplacement physique à divers moment:
  - chargement (allocation statique);
  - exécution (allocation dynamique);

Système de type

- De nombreux langages possèdent :
  - Des types de base : int, char, string, bool
  - Des types pointeurs, tableaux, fonctions
  - La possibilité de définir de nouveaux types produits (structures)
- Un système de types est le plus utile lorsqu'il est inviolable.
  - Le langage C au contraire possède un système de types non sûr (pour de raisons de performances et de simplicité).
  - Java, Caml, Haskell ont des systèmes de types sûrs.
  - Lisp est un langage nontypé.

Structures de contrôle

- Trois structures de contrôle : Composition, Sélection et L'itération
- **Composition :**
  - Est la composition séquentielle de deux ou plusieurs commandes.
  - C'est la structures de contrôle la plus utilisée.
  - Disponible dans chaque Langage de programmation impératif
- **Sélection :**
  - Permet la spécification d'une séquence de commandes par cas.
  - La sélection d'une séquence particulière est basée sur la valeur d'une expression.
  - "If" et "case" sont les commandes les plus utilisées, et les plus représentatives.

Structures de contrôle

- **Itération:**
  - Une commande itérative a un corps qui est exécuté de façon répétitive avec une expression qui détermine quand l'exécution S'arrête.
  - Les plus utilisées:
    - While-do: while condition do body;
    - Repeat-until: repeat body until condition;
    - For-do: for index := lower Bound, upper Bound, step do body;

Les échappement (Escape)

- Le langage machine inclut les instructions qui permettent à n'importe quelle instruction d'être choisie comme prochaine instruction.
- Ils Fournissent des constructions pour donner à des langages de programmation à un niveau élevé une partie de flexibilité (sauts, échappements)
  - Return Exp : Est utilisée en C pour quitter un appel de fonction et pour renvoyer la valeur calculée par la fonction.
  - Exit en ADA/ Break en C: Permet de transférer le contrôle du programme de l'endroit ou la commande exit se trouve vers la première commande, après la boucle la plus interne qui suit le exit.
  - Les exceptions permettent également de remonter brutalement dans la structure de contrôle, mais elles permettent de plus de fournir une valeur exceptionnelle, qu'on peut utiliser ensuite.

Sous Programme, procédures et fonctions

- Une procédure est une abstraction d'une séquence de commandes.
- L'appel de procédure est une référence a l'abstraction de syntaxe
- La syntaxe générale se compose de deux parties:
  - En-tête et corps: Nom (liste des paramètres) { corps }
  - Appel: Nom (liste des arguments)

Plan du cours

- Introdcution
- Variable et assignement
- Système de type
- Structures de contrôle
- Les échappement (Escape)
- Difficultés liées au paradigme impératif
  - Les effets de bord
  - Synonymie (Aliasing)
  - Exception
- Conclusion

Difficultés liés au paradigme impératif

- Le paradigme impératif inclut quelques énoncés qui augmentent la difficultés de compréhension et de manipulation des programmes impératifs. Ces difficultés peuvent être liées aux
  - Effets de bord
  - Synonymies (aliasing)
  - Gestions des exception

Effet de bord en impératif

- En informatique, une fonction est dite à effet de bord si elle modifie un état autre que sa valeur de retour.
- La programmation impérative emploie des effets de bord dans le fonctionnement de ses programmes.
- Les effets de bord rendent souvent le comportement des programmes plus difficiles à comprendre.

Les effets de bord

```
#include <iostream>
using namespace std;

int a;

void f()
{
  a = 2;
}

int main ()
{
  a = 1;
  cout << a << endl;
  f();
  cout << a << endl;
}
```

L'effet de bord de la fonction f est de modifier la valeur de la variable globale a.

Synonymie (Aliasing)

- Deux noms sont des alias s'ils dénotent (partagent) le même objet de données pendant un lancement d'unité
- La synonymie est un autre dispositif des L. P impératifs qui rend des programmes plus durs à comprendre.

Exemple (PASCAL)

```
Procedure confuse (var m, n : Integer );
    begin
      n := 1; n := m + n
    end;
...
i := 5;
confuse(i,i)
```

Les Exceptions

- C’est un événement "anormal" qui survient au cours de l'exécution.
- Les exceptions se produisent lorsque des situations anormales surviennent durant l'exécution d'un programme.
- Lorsqu'une exception se produit, l'exécution normale du programme est interrompue.
  - Les erreurs/exceptions les plus courantes sont probablement l'accès non autorisé à une zone mémoire :
  - Erreur de manipulation de pointeur;
  - Division par zéro (on ne prévoit pas le cas où le diviseur est nul).
  - Des mesures pour différencier l'exécution normale de l'exécution dans un contexte exceptionnel s’avers utile

Conclusion

- La programmation impérative est basée sur le modèle machine de Von Neumann.
- Dans le paradigme impératif est caractérisé par la programmation avec un état et des commandes qui modifient l'état.
- Les difficultés liés aux paradigmes impératif peuvent être résumé comme suit
  - Les effets de bord
  - Les synonymies (aliasing)
  - La gestion des exceptions
- La complexité de raisonner provenant des difficultés de la programmation impérative a été une motivation forte pour fournir des solutions comme les paradigmes fonctionnels et logiques.