
Informatique Générale

 Guillaume Hutzler
Laboratoire IBISC
(Informatique Biologie Intégrative et Systèmes Complexes)
guillaume.hutzler@ibisc.univ-evry.fr
Cours Dokeos 625
<http://www.ens.univ-evry.fr/modx/dokeos.html>

Informatique générale - Introduction aux langages de programmation

Plan et objectifs du cours

- Objectifs du cours
 - Donner une vue d'ensemble de l'informatique
 - du point de vue **historique**
 - du point de vue des **concepts**
 - du point de vue des **techniques**
 - Donner un aperçu des métiers de l'informatique
- Séances
 - 1-2 : Histoire de l'informatique
 - 3-4 : Fondements mathématiques de l'informatique
 - 5-6 : Architecture des ordinateurs et des micro-processeurs
 - 7-8 : Systèmes d'exploitation
 - 9-10 : Langages de programmation
 - 11-12 : Réseaux

Informatique générale - Introduction aux langages de programmation

Informatique Générale Langages de programmation

 Guillaume Hutzler
Laboratoire IBISC
(Informatique Biologie Intégrative et Systèmes Complexes)
guillaume.hutzler@ibisc.univ-evry.fr

Informatique générale - Introduction aux langages de programmation

Le système d'exploitation

- Le **système d'exploitation** (SE, en anglais *Operating System* ou *OS*) est un ensemble de programmes responsables de la liaison entre les ressources matérielles d'un ordinateur et les applications de l'utilisateur (traitement de texte, jeu vidéos, etc.). Il assure le démarrage de l'ordinateur, et fournit aux programmes applicatifs des interfaces standardisées pour les périphériques.
- Typiquement, un Système d'Exploitation est composé :
 - d'un noyau ;
 - de bibliothèques dynamiques ;
 - d'un ensemble d'outils système ;
 - de programmes applicatifs de base.



Informatique générale - Introduction aux langages de programmation

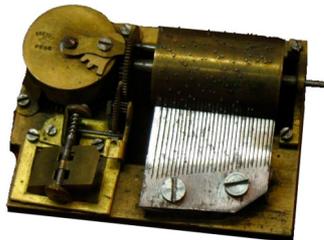
Qu'est-ce qu'un programme?

- Un **programme informatique** est une liste d'ordres indiquant à un **ordinateur** ce qu'il doit faire. Il se présente sous la forme d'une ou plusieurs séquences d'instructions, comportant souvent des données de base, devant être exécutées dans un certain ordre par un **processeur** ou par **processus informatique**
- Un **algorithme** énonce une résolution sous la forme d'une **série d'opérations à effectuer**. La mise en œuvre de l'algorithme consiste en l'écriture de ces opérations dans un **langage de programmation** et constitue alors la brique de base d'un **programme informatique**

Informatique générale - Introduction aux langages de programmation

A. Favre - La boîte à musique (1796)

- évolution des carillons
- cloches et marteaux remplacés par le peigne ou clavier et les picots du rouleau



Informatique générale - Introduction aux langages de programmation

J. M. Jacquard – le métier Jacquard (1801)

- But = limiter le travail des enfants
- Techniques
 - aiguilles de B. Bouchon
 - cartes perforées de J. B. Falcon
 - cylindre de J. de Vaucanson
- Considéré par beaucoup comme l'ancêtre de l'ordinateur
 - cartes perforées encore utilisées dans les années 70 pour programmer les ordinateurs!!!



Informatique générale – Introduction aux langages de programmation

Programmation de la machine analytique

- La machine analytique est programmable
 - grâce aux cartes perforées en entrée
 - un lecteur pour le programme
 - un lecteur pour les données
 - permet à la machine de réaliser
 - des boucles
 - des branchements conditionnels (saut de quelques instructions en avant ou en arrière)
 - instructions contrôlées par le séquenceur
 - cylindre poussant des barres pour réaliser les différentes étapes d'une opération de calcul
 - peut à tout moment indiquer l'étape suivante
 - rotation commandée par un compteur

Informatique générale – Introduction aux langages de programmation

A. Byron, comtesse de Lovelace – l'algorithmique

- Collaboratrice de Babbage
 - chargée de réaliser les programmes de la machine
 - première « programmeuse » de l'histoire
 - invente la technique des sous-programmes



« La machine analytique n'a nullement la prétention de créer quelque chose par elle-même. Elle peut exécuter tout ce que nous saurons lui ordonner d'exécuter [...] Son rôle est de nous aider à effectuer ce que nous savons déjà dominer »

- Euclide – *Les éléments* (300 av. J.C.) – algorithme de calcul du PGCD
- Abou Jafar Muhammad Ibn Mūsa al-Khwarizmi (780–850) – décrit des méthodes de calcul algébrique
- Averroès (1126–1198) – décrit une méthode de raisonnement ou la thèse s'affine étape par étape

Informatique générale – Introduction aux langages de programmation

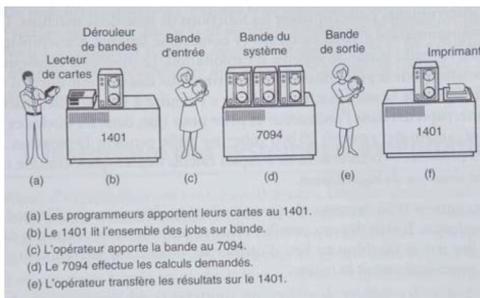
J. Mauchly / J. P. Eckert - ENIAC (1943-1945)

- Moore School (Université de Pennsylvanie)
 - Electronic Numerical Integrator And Computer
 - destiné au calcul de tables balistiques
 - souvent considéré comme le premier ordinateur
- Caractéristiques
 - 30 tonnes / 150 KW / 18000 tubes à vide / 200 KHz
 - programmation par recablage
 - utilise la base 10
 - accumulateurs
- Fonctionne jusqu'en 1952
 - fragile (1 panne / 3 jours)



Informatique générale - Introduction aux langages de programmation

Protocole d'utilisation de la machine



[A. Tanenbaum. Systèmes d'exploitation - 2ème édition. Pearson Education, 2003]

Informatique générale - Introduction aux langages de programmation

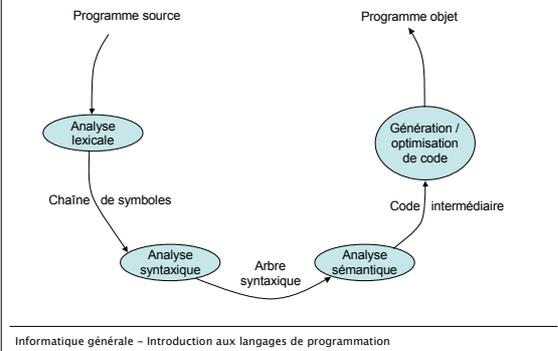
La notion de compilation

- La machine ne comprend que le langage machine

```
0x33 00011010 01011010 11101010 11101010
0x34 01011110 01011010 11000111 11111000
0x35 01011010 01011010 10101010 10101010
0x36 11011110 11101010 10101010 10101010
0x37 11101010 11000111 11000111 11111111
```
- ... mais
 - les humains ont du mal à comprendre le langage machine !
 - le langage machine est spécifique d'un microprocesseur donné
 - absence de portabilité
- développement de langages de haut niveau
 - permettent de s'abstraire des détails de fonctionnement de la machine
 - la compilation correspond à la phase de traduction depuis le langage de haut niveau vers le langage machine

Informatique générale - Introduction aux langages de programmation

Les différentes étapes de la compilation



La notion d'interprète

- outil ayant pour tâche d'analyser, de traduire et d'exécuter un programme écrit dans un langage informatique
- On différencie un programme dit script, d'un programme dit compilé :
 - Un programme script est exécuté à partir du fichier source via un interpréteur de script.
 - Un programme compilé est exécuté à partir d'un bloc en langage machine issu de la traduction du fichier source.
- Le cycle d'un interprète est le suivant :
 - lire et analyser une instruction (ou expression) ;
 - si l'instruction est syntaxiquement correcte, l'exécuter (ou évaluer l'expression) ;
 - passer à l'instruction suivante.

Informatique générale - Introduction aux langages de programmation

Langages compilés ou interprétés

- Avantage des langages interprétés
 - facilité de programmation
 - possible d'exécuter des programmes incomplets
 - portabilité
 - le même programme est exécutable sur n'importe quelle machine où est disponible l'interprète
 - un code compilé n'est exécutable que sur les machines compatibles avec celle où il a été produit
- Inconvénients
 - relative lenteur par rapport au code compilé

Informatique générale - Introduction aux langages de programmation

Exemples de programmation déclarative (1)

- **HTML** (Hyper Text Markup Language)
 - utilisé pour la description de pages Web
 - une page Web écrite en HTML décrit le contenu de la page, mais pas explicitement la manière de calculer l'aspect et la position des différents objets de la page
- **SQL** (Structured Query Language)
 - 1979-2003
 - utilisé pour manipuler et interroger des bases de données relationnelles
 - Exemple

```
SELECT (name, service) FROM employees
WHERE (statut='stagiaire')
GROUP BY (name,service)
ORDER BY name;
```

Informatique générale - Introduction aux langages de programmation

Exemples de programmation déclarative (2)

- **Prolog** (Programmation logique)
 - ensemble de
 - faits élémentaires
 - règles de logique
 - les faits et règles sont exploités par un *démonstrateur de théorème* ou *moteur d'inférence*, en réaction à une question ou requête
 - Exemple de programme

```
frère_ou_sœur(X,Y) :- parent(Z,X), parent(Z,Y), X \= Y.
père(X,Y) :- parent(X,Y), mâle(X).
mère(X,Y) :- parent(X,Y), femelle(X).
parent(X,Y) :- père(X,Y).
parent(X,Y) :- mère(X,Y).
mère(trude, sally).
père(tom, sally).
père(tom, erica).
père(mike, tom).
mâle(tom).
femelle(trude).
mâle(mike).

?- frère_ou_sœur(sally, erica)
oui
```

Informatique générale - Introduction aux langages de programmation

Exemples de programmation déclarative (3)

- La **programmation fonctionnelle**
 - traite le calcul comme l'évaluation de fonctions mathématiques
- Exemple de programme **Lisp**

```
(defun factorial (n)
  " Calcule la factorielle de l'entier n. "
  (if (<= n 1)
      1
      (* n (factorial (- n 1)))))
```
- Autres langages fonctionnels
 - ML, Haskell, Ocaml, Erlang, XSLT, ...

Informatique générale - Introduction aux langages de programmation

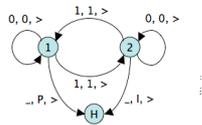
La programmation impérative

- langages théoriques
 - Machine de Turing
- Langages de bas niveau
 - Langage machine
 - Assembleur
- Langages de haut niveau
 - Programmation structurée et procédurale
 - Programmation orientée objet

Informatique générale - Introduction aux langages de programmation

La machine de Turing (1936)

- Comprend
 - un bande de longueur illimitée constituée de cellules
 - un contrôleur
 - comporte un nombre limité d'états, dont un état spécial, l'état de départ, qui est l'état initial de la machine avant son exécution
 - peut lire et écrire des symboles sur la bande
 - agit en fonction de l'état courant et du symbole lu en activant des règles (dont l'ensemble constitue le programme) :
 - réécrire le symbole courant
 - changer d'état
 - se déplacer à gauche ou à droite sur la bande



Informatique générale - Introduction aux langages de programmation

Les langages machine

- langage natif du processeur
 - correspond aux instructions chargées depuis la mémoire dans le microprocesseur et directement exécutables
 - spécifique d'un microprocesseur
 - les instructions sont les opérations de base du microprocesseur
 - chercher une valeur dans la mémoire pour charger un registre
 - additionner deux registres
 - etc.
- programme =
 - longue suite d'instructions exécutées en séquence
 - sauf en cas de saut (conditionnel ou non)
- exemple d'instruction MIPS
 - ajouter les registres 1 et 2 et placer le résultat dans le registre 6

```
[ op | rs | rt | rd | shamt | fonct ]
0   1   2   6   0   32   décimal
000000 00001 00010 00110 00000 100000   binaire
```

Informatique générale - Introduction aux langages de programmation

Les langages d'assemblage (ou assembleurs)

- Principes
 - créés pour faciliter le travail des programmeurs
 - correspondance 1 à 1 entre les instructions du langage machine et le langage d'assemblage correspondant
 - mots du langage machine représentés par des symboles appelés *mnémoniques* (faciles à retenir)
 - on peut traduire le code dans les deux sens sans perte d'information
- Exemple : processeur x86
 - en langage machine : 10110000 01100001
 - en langage d'assemblage : mov %al, \$0x61
 - en langage naturel : charger la valeur hexadécimale 61 dans le registre 'AL'

Informatique générale - Introduction aux langages de programmation

Fortran (1957-2003)

- Principes
 - créé pour accélérer l'écriture de programme
 - réduit le nombre d'instructions nécessaires d'un facteur 20
 - dédié aux applications scientifiques
- Exemple de programme

```
C AREA OF A TRIANGLE WITH A STANDARD SQUARE ROOT FUNCTION
C INPUT - CARD READER UNIT 5, INTEGER INPUT
C OUTPUT - LINE PRINTER UNIT 6, REAL OUTPUT
C INPUT ERROR DISPLAY ERROR OUTPUT CODE 1 IN JOB CONTROL LISTING
      INTEGER A,B,C
      READ(5,501) A,B,C
501  FORMAT(3I5)
      IF(A.EQ.0 .OR. B.EQ.0 .OR. C.EQ.0) STOP 1
      S = (A + B + C) / 2.0
      AREA = SQRT( S * (S - A) * (S - B) * (S - C) / 2.0)
      WRITE(6,601) A,B,C,AREA
601  FORMAT(4H A= ,15,5H B= ,15,5H C= ,15,5H AREA= ,F10.2,12HSQUARE UNITS)
      STOP
      END
```

Informatique générale - Introduction aux langages de programmation

Programmation « spaghetti »

- Exemple de programme spaghetti

```
0  i = 0
20 i = i + 1
30 IF i <> 11 THEN GOTO 80
40 IF i = 11 THEN GOTO 60
50 GOTO 20
60 PRINT "Programme terminé."
70 END
80 PRINT i & " au carré = " & i * i
90 GOTO 20
```
- Version améliorée

```
10 FOR i = 1 TO 10
20 PRINT i & " au carré = " & i * i
30 NEXT i
40 PRINT "Programme terminé."
50 END
```

Informatique générale - Introduction aux langages de programmation

Programmation structurée

- Objectifs
 - rendre les programmes
 - plus lisibles
 - plus faciles à modifier
 - supprimer les instructions `GOTO`
- Principes = techniques et méthodologies de structuration des programmes
 - structuration des données
 - structuration des instructions
- Structures de contrôle de bas niveau
 - concaténation
 - sélection
 - répétitions

Informatique générale - Introduction aux langages de programmation

Structures de contrôle de bas niveau (1)

- **Concaténation** = séquence d'instructions exécutées dans l'ordre

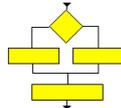
```
0 i = 0
20 i = i + 1
```



- **Sélection** = exécution d'une instruction parmi n en fonction de l'état du programme

```
30 IF i <> 11 THEN GOTO 80
```

- formes courantes
 - `if ... then ... else ... endif`
 - `switch / case`



Informatique générale - Introduction aux langages de programmation

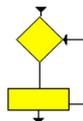
Structures de contrôle de bas niveau (2)

- **Répétitions** = exécution d'un ensemble d'instructions jusqu'à ce que

- le programme atteigne un certain état
- tous les éléments d'un ensemble soient traités

```
10 FOR i = 1 TO 10
20 PRINT i & " au carré = " & i * i
30 NEXT i
```

- formes courantes
 - `repeat ... until` ou `do ... until`
 - `for`
 - `while`



Informatique générale - Introduction aux langages de programmation

Programmation procédurale

- paradigme de programmation basé sur la notion d'appel de procédures
- une procédure (appelée aussi routine, méthode, fonction)
 - contient une succession d'étapes de calcul à exécuter
 - peut être appelée à tout moment pendant l'exécution du programme
 - y compris par d'autres procédures
 - y compris par elle-même (procédures récursives)

Informatique générale - Introduction aux langages de programmation

Intérêts de la programmation procédurale

- Intérêts de la programmation procédurale
 - possibilité de réutiliser le même code à différents endroits d'un programme sans avoir à le recopier
 - en cas de modification de ce code, évite d'avoir à rechercher tous les endroits du programme où ce code est utilisé
 - améliorer la lisibilité du programme
 - permettre une programmation modulaire et structurée

Informatique générale - Introduction aux langages de programmation

Exemples de langages procéduraux

- Ada
- ALGOL
- BASIC
- C
- COBOL
- Fortran
- Maple
- Pascal
- etc.

Informatique générale - Introduction aux langages de programmation

Exemple de programmation procédurale en Logo

• Logo & les tortues

- langage créé en 1967 par W. Feurzeig et S. Papert
- objectif éducatif
 - simplicité du langage
 - feedback visuel permettant le débogage
 - tortue robotisée (*Irving*)

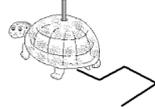
```
to square
  repeat 4 [forward 50 right 90 ]
end
```



```
to flower
  repeat 36 [right 10 square ]
end
```



```
to spiral :size :angle
  if :size > 100 [stop ]
  forward :size
  right :angle
  spiral :size + 2 :angle
end
```



Informatique générale - Introduction aux langages de programmation

La programmation orientée objet

• Principes

- utilise les « objets » et leurs interactions
- basés sur les concepts d'héritage, de polymorphisme, d'encapsulation

• Concepts

- **classe** : définit les caractéristiques abstraites d'un objet
 - ses caractéristiques = ses **attributs** ou champs
 - ses comportements = ses **méthodes**
- **objet** : une instance particulière d'une classe
- **héritage** : une sous-classe est une version spécialisée d'une classe, qui hérite des attributs et comportements de la classe parent et y ajoute les siens
- **encapsulation** : les caractéristiques internes de l'objet sont masquées et ne sont accessibles qu'au travers de méthodes de manipulation de l'objet
- **polymorphisme** : une même méthode sera exécutée différemment en fonction du type de l'objet qui l'exécute

Informatique générale - Introduction aux langages de programmation

Exemple de code objet

```
class Forme {
  public :
  virtual float Aire() = 0;
};

class Carre : public Forme {
  public :
  virtual float Aire() { return m_cote*m_cote; }
  private :
  float m_cote;
};

class Cercle : public Forme {
  public :
  virtual float Aire() { return 3.1415926535*m_rayon*m_rayon; }
  private :
  float m_rayon;
};
```

Informatique générale - Introduction aux langages de programmation

Exemples de langages objets

- Simula
- Smalltalk
- C++
- Java
- OCaml
- Python
- etc.

Informatique générale - Introduction aux langages de programmation
