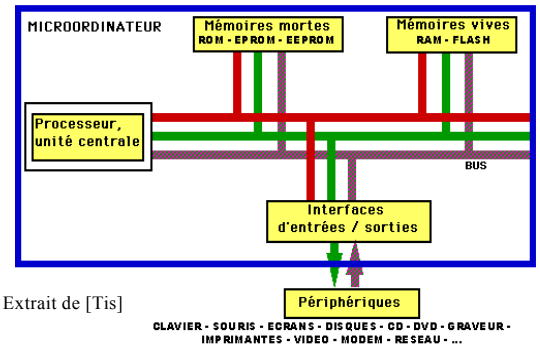
Chapitre 1

Niveaux de description

de l'élégance du langage

Olivier Raynaud Université Blaise Pascal

Base conceptuelle d'un ordinateur

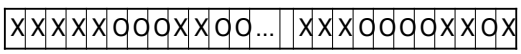


Extrait de [Tis]

La mémoire

❑ *Une décomposition en mots et en bits.*

- La mémoire est divisée en parties physiques appelées mots (par exemple 65 536 mots pour une mémoire).
- Un mot se divise en bits (la taille d'un mot correspond à la taille d'un registre ou du bus)



Les bits sont des contacts magnétiques qui peuvent être dans l'une ou l'autre position.

Interprétation

❑ *Instructions et données*

- ✓ **Mémoire**: Les mots de la mémoire contiennent les données à traiter ou les instructions pour traiter ces données.

La première partie du mot contient le nom du type de l'instruction à exécuter.

La seconde partie contient l'adresse numérique d'un mot (ou des mots) sur lequel exécuter l'instruction.

- ✓ **L' unité centrale** dispose d'un pointeur spécial (le registre appelé compteur ordinal ou IP – PC en français) qui désigne le prochain mot à être interprété comme une instruction.

Exemple

ADD AX, *1983*
MOV AX, *1982*
PUSH AX

Espace de stockage

❑ *Vision conceptuelle de la mémoire*

Un espace de stockage est une collection de cellules.
1.Chaque cellule a un statut courant: alloué ou non alloué
2.Chaque cellule allouée a un contenu courant qui est soit une valeur stockée soit une valeur indéfinie.

Considérons la déclaration suivante en Pascal : **var** n : integer

Une cellule non-allouée devient allouée et son contenu est indéfini, n dénote cette cellule.

Nous pouvons voir chaque cellule allouée comme une boite contenant la valeur d'une variable primitive ou un indéfini « ? « .

Variable

❑ *Variable simple et variable composée*

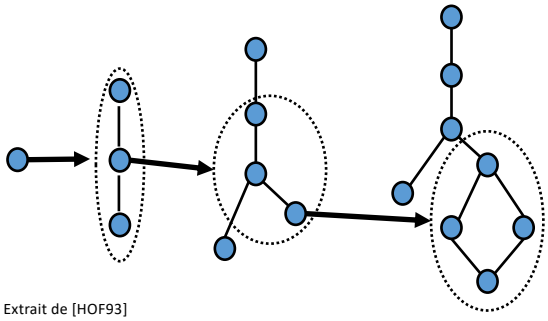
Définition : une variable est un objet qui contient une valeur, cette valeur sera inspectée ou mise à jour aussi souvent que désirée.

- ✓ On distinguera les variables simples des variables de type composé constituées de composants pouvant être inspectés de manière sélective.

```
type  Mois  =  (jan, fev, mar,...,dec)
      Date = record  m : Mois;   j : 1..31  end;
var   leJour  : Date;
...   leJour.j  := 23;  leJour.m  := fev
```

leJour leJour.m
leJour.j

Réunitarisation



Extrait de [HOF93]

Langage

❑ *Caractéristiques propres*

Un langage doit être universel (tout problème doit avoir une solution qui peut être programmé dans le langage);

Un langage doit être le plus naturel possible;

Un langage doit être implémentable sur un ordinateur.

❑ *Syntaxe et sémantique*

✓ **La syntaxe** concerne la forme du programme, la façon dont les variables, les expressions et les instructions sont disposées ensemble pour former un programme.

✓ **La sémantique** concerne le sens à donner au programme, son comportement lors de son exécution.

Langage machine

❑ *Langage natif*

✓ **Seul langage** d'un processeur qu'il puisse traiter. Il est composé d'instructions et de données à traiter codées en binaire.

Le **langage machine**, ou **code machine**, est la suite de bits qui est interprétée par le processeur d'un ordinateur exécutant un programme.

Wikipédia

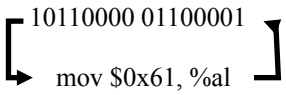
Langage d'assemblage

✓ **Le langage d'assemblage** est situé au dessus du langage machine dans la hiérarchie des langages.

Un **langage d'assemblage** ou **langage assembleur** est un langage bas niveau qui représente le langage machine sous une forme lisible par un humain.

Wikipédia

Il existe une correspondance entre les instructions en langage d'assemblage et les instructions en langage machine.



Un morceau d'ADN

✓ **Morceau** de la séquence de base du chromosome du bactériophage PhiX174.

tcgcgcgatctttgagctaattagagtaaattaatccaatctttgacccaa
atctctgctggatcctctggtatttcatgttggatgacgtcaatttctaatat
ttcacccaaccgttgagcaccttgtgcgatcaattgttgatccagttttatg
attgcaccgcagaaagtgtcatatctgagctgcctaaaccaaccgcccca
aagcgtacttgggataaatcaggcttttgtgatctgttctaataatggctgc
aagttatcaggtagatccccggcaccatgagtggatgtcacgattaacca
caggccattcagcgtaagttcgtccaactctgggccagaagttttctgtag
aaaacccagcttcttctaatttatccgctaaatgttcagcaacatattcagc

Les programmes Assembleur

✓ **Les assembleurs** sont apparus comme le premier outil permettant au programmeur de prendre du recul par rapport au code objet et de se consacrer à la programmation elle même.

Un **assembleur** est un programme qui traduit un programme écrit en langage d'assemblage — essentiellement, une représentation mnémonique du langage machine — en code objet.

Wikipédia

Les langages de compilation

❑ *Quelques réflexions*

✓ **Modèles récurrents :** Modèles fondamentaux lorsque l'on essaie de formuler des algorithmes;

✓ **Unités de haut niveau :** Les programmes sont constitués d'unités de haut niveau indépendantes;

Un **langage de compilation** - ou de programmation - est une notation conventionnelle destinée à formuler des programmes informatiques.

Wikipédia

Les trois niveaux de description

```
7f45 4c46 0201 0100 0000 0000 0000 0000
0200 3e00 0100 0000 3004 4000 0000 0000
4000 0000 0000 0000 e019 0000 0000 0000
0000 0000 4000 3800 0900 4000 1f00 1c00
0600 0000 0500 0000 4000 0000 0000 0000
4000 4000 0000 0000 4000 4000 0000 0000
f801 0000 0000 0000 f801 0000 0000 0000
0800 0000 0000 0000 0300 0000 0400 0000
3802 0000 0000 0000 3802 4000 0000 0000
3802 4000 0000 0000 1c00 0000 0000 0000
1c00 0000 0000 0000 0100 0000 0000 0000
0100 0000 0500 0000 0000 0000 0000 0000
0000 4000 0000 0000 0000 4000 0000 0000
fc06 0000 0000 0000 fc06 0000 0000 0000
0000 2000 0000 0000 0100 0000 0600 0000
100e 0000 0000 0000 100e 6000 0000 0000
100e 6000 0000 0000 2802 0000 0000 0000
3002 0000 0000 0000 0000 2000 0000 0000
0200 0000 0600 0000 280e 0000 0000 0000
280e 6000 0000 0000 280e 6000 0000 0000
d001 0000 0000 0000 d001 0000 0000 0000
0800 0000 0000 0000 0400 0000 0400 0000
5402 0000 0000 0000 5402 4000 0000 0000
5402 4000 0000 0000 4400 0000 0000 0000
4400 0000 0000 0000 0400 0000 0000 0000
```

```
#include <stdio.h>

int main()
{
  printf("hello world");
  return 0;
}
```

```
        .file   "test.c"
        .section        .rodata
.LC0:
        .string "hello world"
        .text
        .globl  main
        .type   main, @function
main:
.LFB0:
        .cfi_startproc
        pushq   %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq    %rsp, %rbp
        .cfi_def_cfa_register 6
        movl    $.LC0, %edi
        movl    $0, %eax
        call    printf
        movl    $0, %eax
        popq    %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
.LFE0:
        .size   main, .-main
        .ident  "GCC: (Ubuntu 5.4.0-6ubuntu1~16.04.4) 5.4.0 20160609"
        .section        .note.GNU-stack,"",@progbits
```

Wikipédia

Compilateurs

❑ *Programme de transformation*

✓ **Le code source** est écrit dans un langage de programmation (langage source), il est de haut niveau d'abstraction et facilement compréhensible par un humain.

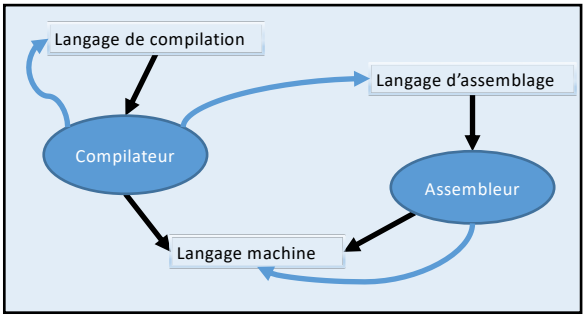
✓ **Le code objet** est écrit dans un langage de plus bas niveau (langage cible) tel un langage d'assemblage.

Un **compilateur** est un programme qui transforme un code source en code objet. De tel programme ont été écrit à partir des années 50.

Wikipédia

Question : *Dans quel langage sont écrits ces compilateurs?*

Amorçage



Extrait de [HOF93]

Interpréteur

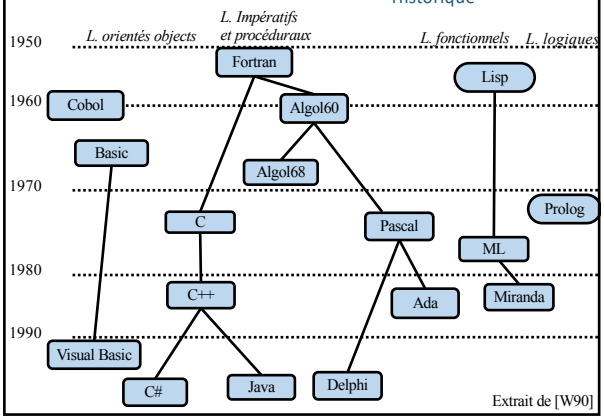
- *Programme de script*

Un **interpréteur** est un programme ayant pour tâche d'analyser, de traduire et d'exécuter les programmes écrits dans un langage informatique.

Wikipédia

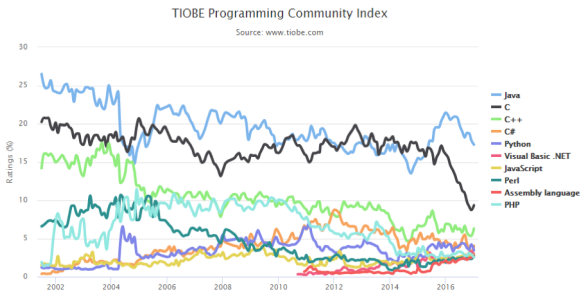
- *Cycle des opérations*
  - ✓ **Lire** et **analyser** une instruction ou une expression;
  - ✓ Si l'instruction ou l'expression est syntaxiquement correcte, l'**exécuter** ou l'**évaluer**;
  - ✓ passer à l'instruction suivante.

Historique



Extrait de [W90]

Communauté de programmeurs



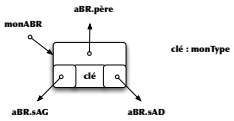
Langage de Description d'Algorithme
(LDA)

- *Langage de pseudo code*

Le **LDA** est un langage de rédaction de routines de calcul, de manipulation et de gestion de la mémoire. Sous le paradigme objet, ce sera le langage utilisé pour la rédaction des algorithmes.

Algorithme 1: $aBR.parcours-infixe()$ (Récursif)
Complexité : $\mathcal{O}(n)$

Données : $self$:aBR
Résultat : néant (Un affichage à l'écran de chacune des clés de $self$.)
début
  si $(self.sAG)$ **alors**
    $self.sAG.parcours-infixe()$;
  fin
  $afficher(self.clef())$;
  si $(self.sAD)$ **alors**
    $self.sAD.parcours-infixe()$;
  fin
fin



Calculabilité et algorithmique

Une fonction f est ***calculable*** s'il existe un procédé systématique permettant à partir de la valeur « x », par une série de manipulations précises, de connaître «f(x)».

En février 34, A.Church soulève la question suivante:

- ✓ ***Question :*** *Quel est l'ensemble d'outils, le kit d'opérations, nécessaire pour calculer les valeurs des fonctions calculables?*

- ✓ Les fonctions calculables avec le Kit algorithmique (L.D.A.) sont par définition les fonctions *programmables.*

Thèse : Toute fonction calculable est programmable et réciproquement.

Alonzo Church

Pour résumer

❑ *Quelques points essentiels de l'exposé*

- ✓ **Machine :** Nous avons décrit un micro ordinateur comme composé d'une mémoire, d'un C.P.U. et d'un ensemble d'entrée/sortie. La fonction d'un ordinateur est d'exécuter des instructions sur des données.
- ✓ **Mémoire :** La mémoire d'un ordinateur peut être vu comme un ensemble de mots, composés de bits. D'un point de vue symbolique la mémoire est un espace de stockage composés de cases (allouée, vide ou pleine).
- ✓ **Langage :** Nous avons décris les propriétés des langages machine, d'assemblage et de compilation. Nous avons rappeler les différents paradigmes de programmation et le nom des langages manipulés par la communauté.
- ✓ **Langage pseudo code (LDA):** Le langage décrit permet la gestion de la mémoire à un niveau abstrait adapté et la modélisation du calcul sous un format objet. Le LDA distingue clairement le concept de fonction du concept de procédure.