ALM : Architectures logicielles et Matérielles

- Pascal.Sicard@imag.fr
- Bureau 313 Bâtiment D ENSIMAG

- Organisation de l'enseignement d'ALM
  - 10 * 1h30 heures de cours + 8* 3 Travaux dirigés de 2h
  - 2 séances de SOUTIEN sur machine en fin de semestre
  - Préparation de travaux pratiques en TD
  - Travail personnel sur machine en libre service
  - Comptes rendus (note de contrôle continu) par binôme
  - Note finale ALM1 : Examen coefficient 2 , Contrôle Continu 1

Organisation de l'enseignement

- **Partie hard (matériel):** éléments matériels nécessaire à la conception de processeurs et d'ordinateurs
- **Partie soft (logiciel):** Langage d'Assemblage, introduction à la traduction de programmes en langage de haut niveau permettant leurs exécutions par les circuits
- 1 TD "hard", 1 TD "soft", 1 TD "général" («hard bis»)
- **Attention à l'emploi du temps qui n'est pas "régulier" (supports de TDs)**
- **Contrôle continu:**
  - 3 comptes rendu de TPs Soft
  - 3 comptes rendu TPs Hard
  - Un compte rendu à rendre par binôme sur papier à l'enseignant concerné

Bibliographie et communication

- **Bibliographie**
  - P. Amblard, JC Fernandez, F. Lagnier, F. Maraninchi, P. Sicard, P. Waille Architectures logicielles et matérielles. Editions DUNOD 2000
  - En bibliothèque ou en ligne sur le WEB (dans le Moodle)
- **Outil pédagogique Moodle à l'UFR IMAG:**
  - http://imag-moodle.e.ujf-grenoble.fr

Objectifs

- Comprendre le fonctionnement des circuits composants les ordinateurs
- Comprendre comment un programme est exécuté par un ordinateur
- Notions indispensables à tout informaticien polyvalent
  - En particulier quand on «touche»
    - au système d'exploitation
    - aux périphériques et entrée/sortie
    - programmation «haute performance» (contrainte mémoire, temps réel ...)
  - Bases de la conception de circuit....
- 2ème semestre : Interface Logiciel/matériel et système (ALM2)
  - Processeur et son entourage, les entrées/sorties, les bases du système d'exploitation

Chapitre 1 : c'est quoi un ordinateur

Introduction

- Ethymologie : adjectif existant depuis longtemps :"Dieu qui met de l'ordre dans le monde"
- Computer : calculateur trop restrictif pour les premiers fabricants français
- Un ordinateur: une machine qui traite des informations
- Traitement : enchaînement "programmé" de calculs sur une représentation particulière de l'information
- On parle de traitement "numérique"
- Changements possibles des programmes pour effectuer des traitements à volonté

Ordinateurs





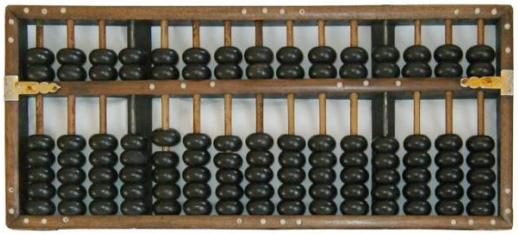

Brève historique

http://www.histoire-informatique.org

- La préhistoire

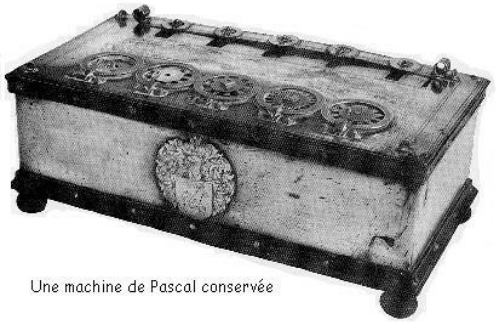
-3000 an AvJC : **Boulier**

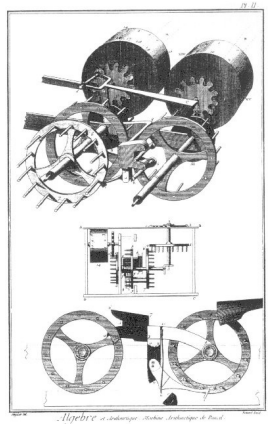
- Appareil de calcul qui permet d'avoir la représentation du nombre (mémoire) et d'effectuer une opération sur ce nombre (calcul)



Première machine à calculer en bois et métal

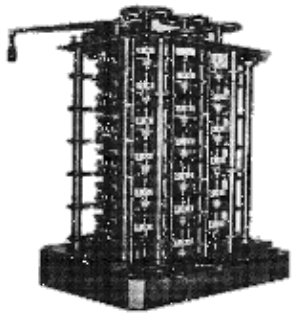
- 17e siècle premier calcul automatique :
  - Machine de Pascal : roues crantées pour effectuer des additions et des soustractions.


Une machine de Pascal conservée



Historique 19 ème

- 1820: **Machine à différences** de C. BABBAGE
- jamais finalisé, 25000 pièces, système décimal



- 1833: **Machine analytique** de C. BABBAGE
  - **utilisé à l'aide de cartes *opérations* , de cartes *de variables* et de cartes *nombres*.**
  - **Ancêtre de l'ordinateur**

Historique début 20 ème siècle
Du mécanique à l'électrique

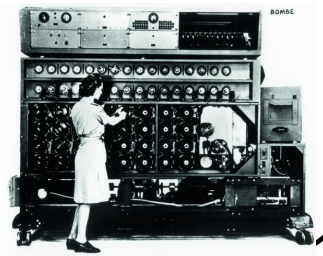
- Pas encore vraiment des ordinateurs: spécialisés dans des calculs particuliers (multiplication, décalage ...): 1 seul programme
- **1938; *Z1:*** premier calculateur d'abord mécanique puis à relais électriques par K. ZUSE
- **1938; Thèse de Shannon :** Relation logique de Boole et circuit électrique - Invention du Bit (BInary digiT)
- **1939: *ABC*** (J. ATANASOFF)**:** premier additionneur 16 bits électrique binaire (l'information est représentée par des 0 et des 1)

Vers une machine algorithmique universelle

- **1938;** Thèse de Turing-Church sur la calculabilité
- Machine de Turing abstraite permettant d'exécuter n'importe quel algorithme

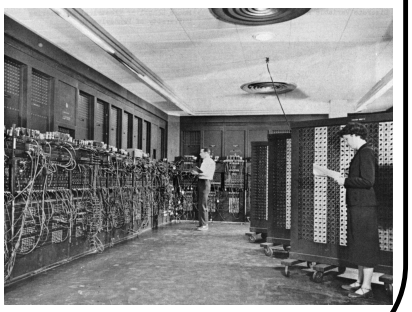
- **Machine Enigma (1920) :** chiffrement de texte, utilisé pendant la guerre 39-45

- **1940;** A. Turing - construction de la "bombe" permettant de décrypter les messages encodés avec Enigma
  - Secret défense jusqu'en 1975



Changement des programmes par câblage

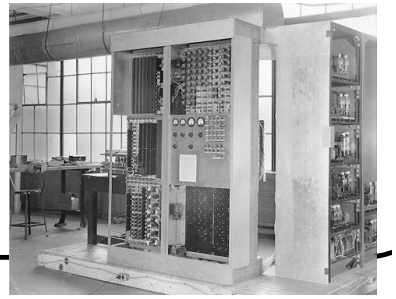
- **1941; Zuse3** à base de tube à vide- Binaire, mémorisation des variables, premiers programmes
- **1943**; **Colossus** (évolution de la bombe de Turing) Première utilisation de l'électronique binaire
- **1943; ASCC** ou **Harvard MARK 1** par H. AIKEN :
  - en collaboration avec IBM , à base de milliers de relais, 5 tonnes, refroidissement par glace,
  - 3 opérations sur 23 bits par seconde,
  - pas d'**instruction conditionnelle**



- **1946**; **ENIAC** (Electronic Numerical Integrator Analyser and Computer)
  - Turing-complet
  - Codage décimal

Premiers ordinateurs
“On peut mémoriser les programmes”

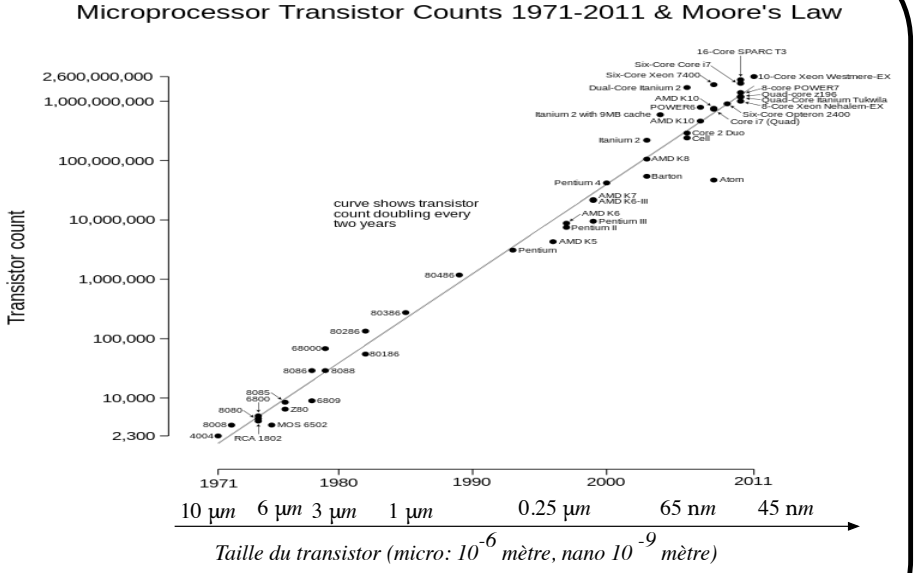
- **1948**; Machine de Von Neuman (EDVAC: Electronic Discrete Variable Automatic Computer)
- Architecture identique dans les ordinateurs d’aujourd’hui
- Mémoire pouvant contenir un programme et ses données
- 2096 mots de 40 bits
- Une unité centrale permettant de réaliser des calculs
- 1 seul registre accumulateur
- Instructions sur 20 bits



Jusqu’aux premiers ordinateurs personnels

- **1955** : Premiers ordinateurs à transistors (1948 : invention du transistor)
- **1971** : Premiers ordinateurs personnels (2000 transistors sur un circuit)
- **1981** : Premiers PC
- **1982** : A la faculté de Grenoble : Un ordinateur (système Multics) mémorisation sur bande, programmation en assembleur sur cartes perforées (1 carte = une instruction). Mais aussi en langage de haut niveau.
- **1990** : RAM en kilo octets, Disque dur en Méga octets, fréquence horloge en Méga hertz
- **2000** : fois 1000 en 10 ans (Loi de Moore (1965) double tous les ans)

Loi de Moore (source wikipedia)



Taille du transistor (micro: 10^{-6} mètre, nano 10^{-9} mètre)

2014 : 5 milliards de transitors, 20 nanom

Quelques chiffres de comparaison des premiers microprocesseurs à aujourd’hui

- Nombre de transistors * 2 millions (2300 à 5 milliards)
- Fréquence de l’horloge * 6000 (740 kHz à 5 gHz)
- Taille des transistors / 500 (10 micro à 20 nano)
- Taille des registres * 16 (4 à 64 bits)

Mémoires:

- Taille disque dur * 1 million (Méga à Téra)
- Taille RAM * 1 million (Kilo à Giga)

La représentation de l'information

- Une information: nombre, caractère, texte, image, son, couleur...
- Dans une machine ces informations sont codées
  - Différents codes possibles:
    - un entier: des lignes de boules déplacées, des roues crantées positionnées, du courant électrique...
    - Du son: une gravure sur du vinyle, du courant électrique, vibration d'une membrane...
- **Acquisition de l'information :** Clavier, micro, caméra, réseau
- **Restitution de l'information :** écran, haut-parleur, commande de robots, réseau

Codage de l'information dans un ordinateur

- Deux niveaux de tension électriques (0 volt et 5 volts)
- Circuits plus "simple" à concevoir
- Codage binaire : 1 et 0
- **Toute information est représentée par une suite de 1 ou 0**
- On parle de **codage digital** (circuit digital)
  - Vient de digit : doigt
- **Bit (contraction de Binary Digit)** code élémentaire : 1 ou 0 **(Chiffre binaire)**
- **Octet** : 8 bits
  - S'écrit souvent en **base 16 (chiffre hexadécimal)**
  - 4 bits = 1 chiffre hexadécimal (0 à 9, A, B, C, D, E, F)
  - Exemple : 0011 1010 (base 2)= 3A (base 16)

Quantité d'information

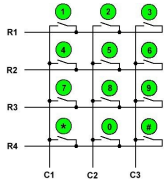
- Sur 8 bits : 256 valeurs. Sur n bits : 2^n valeurs
- On parle de kilo/méga/giga/téra octets (ou bits)
- Par habitude les quantités de stockage sont en octet (Byte); Les débits en bit
- On parle de Poids fort -Poids faible
  - Exemple : 1000 1011 :
    - 1000 : 4 bits de poids forts,
    - 1011 : 4 bits de poids faible
- Pour les quantités de stockage les ordres de grandeurs font référence au puissance de 2 :
  - Kilo : $2^{10}=1024 \neq 10^3 = 1000$
  - Méga : $2^{20}=1\,048\,576 \neq 10^6$
  - Giga : : $2^{30}=1\,073\,741\,824 \neq 10^9$
  - Téra : : $2^{40}=1\,099\,511\,627\,776 \neq 10^{12}$

Interprétation de l'information binaire

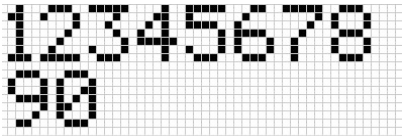
- Il peut y avoir différents codages numériques de la même information
- Exemple: 0 1 0 0 0 0 0 1 peut représenter l'entier 65 (en base 2), un caractère ("A" en code ASCII), une couleur, une instruction machine ....
- **C'est une question d'interprétation**
- **Exemple courant de mauvaise interprétation:**
  - **Un fichier contenant des instructions machines est interprété comme des codes ASCII**

(texte):<FE><ED><FA><CE>^@^@^@^R^@^@^@^@^@^@^@^B^@^@^@^K^@^@^D<F0>^@^@^@<
^@8__PAGEZERO^@^@^@^@^@^@^@^@^@^@^@^@^P^@^@^@^@^@^@^@^@^@^@^@^@^@^@^
^@^@^@__picsymbol_stub__TEXT^@^@^@^@^@^@^@^@^@^@^@^@^@3<D4>^@^@^Fx^@^@#<D4>^
g^@^@^@^@^@^@^@^@__TEXT^@^@^@^@^@^@^@^@^@^@^@^@^@:L^@^@^E<B4>^@^@*L^@^@^@
^B^@^@^@^@

Différents codages d'une même information



- Un caractère tapé au clavier
  - Une touche appuyée -> contact électrique
    - Génération du code ASCII représentant ce caractère (Ex chiffre 2: 00110010)
    - Traitement par l'ordinateur de cette donnée, éventuellement transformation du code ASCII en valeur en base 2 (ou en complément à 2) si le caractère est un chiffre (Ex : 0000 0010)
- Affichage à l'écran :
  - Transformation du code ASCII en vecteurs de bits qui représentent les coordonnées des pixels à allumer à l'écran
  - Envoi d'électrons sur l'écran



Analogique / numérique

- **Représentation analogique :**
  - Sens physique, continu
  - Exemple : disque 33 tours vibration plus ou moins rapide d'un diamant qui va donner une variation de tension électrique
- **Représentation numérique ;**
  - Disque CD : trou au laser représentant des 1 (trou) et des 0 (pas de trou)
- Dans le cas d'un CD, l'information (le son ici) n'a pas besoin de changer de codage en entrée, elle peut être traitée directement
- Par contre à la restitution, il faudra la rendre analogique (tension électrique sur les hauts parleurs).

Codages binaires des caractères

- Code **ASCII** (American Standard Code for Information Interchange) sur 7 bits
- 1 bit pour les accents, non normalisé: problème classique du passage des caractères accentués entre différents types de d'ordinateur
- Exemple:
  - Code ASCII de a = 97 (base 10) = 61 en héxadécimal
  - Code ASCII de b= 98 (base 10)
- ***man ascii (sous unix)***

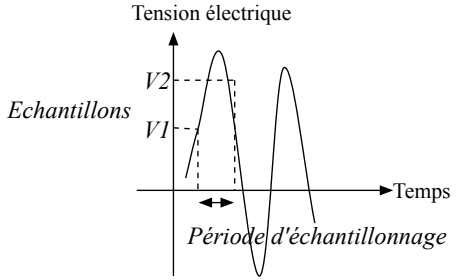
Codages des entiers

- Entiers naturels : **Base 2**
- Entiers relatifs : **complément à 2** (voir en TD) qui permet d'utiliser les mêmes opérations (et circuits) que pour la représentation en base 2
- Taille dépend du matériel, on parle de processeur 32 bits, 64 bits : taille des opérandes
- Taille supérieur possible mais opérations "décomposées" (Poids faibles - poids fort)

Codages des réels

- **Virgule flottante** : signe, mantisse, exposant
- Exemple en base 10:
  - 10,3 : $0{,}103.10^3$: signe +, mantisse entre 0 et 1: 0,103; exposant: 3
- Exemple en base 2 : norme 754 sur 32 bits
  - Signe s: 1 bit, mantisse sur 23 bits, exposant sur 8 bits
- Opération en virgule flottante:
  - Addition/soustraction:
    - Cadrage avant l'opération: même exposant
    - Addition/soustraction des mantisses
    - Post normalisation : premier chiffre après la virgule de l'exposant non nul
- Problème de **précision**: (normalisation) choix entre la taille de la mantisse et de l'exposant

Codage du son



- La qualité de restitution dépend de la fréquence d'échantillonnage et du nombre de bits pour coder la valeur des échantillons
- Exemple:
  - Qualité téléphone: 8000 fois par seconde, échantillon sur 1 octet -> 64 kilobits/s
  - CD Audio: 44100 Hertz, échantillon sur 2 octets : 700 kilobits/s (stéréo * 2)
- Compression possible :
  - Exemple de principe de compression: codage des différences entre les échantillons
  - exemple MP3: Suppression des fréquences non audibles pour l'oreille humaine: gain variable de 10 à 30 fois

Codage des images

- 1 pixel: point sur l'écran
- 1 image qualité télévision HD: 1920*1080 = 2 Méga-pixels
- 1 pixel: une couleur: 3 niveaux rouge/vert/bleu ou valeur dans une palette prédéterminée
  - Exemple: 1 octet par couleur: 3 octets : $\mathbf{2^{24}}$ nuances de couleurs (16 millions)
- 1 image couleur: 3*2= 6 méga octets pour une image
- On parle de format bit map (BMP) ou encore GIF, TIFF
- Compression possible : exemple JPEG: gain jusqu'à 50 fois
- Logiciel de compression/décompression

Codage de la vidéo

- **Vidéo : suite d'images + son**
  - film : 25 images /secondes
  - 6 mégaoctets * 25 = 150 mégaoctets / seconde !
- **Compression MPEG (***Moving Picture Experts Group)*
  - *Succession de différentes normes : MPEG 1, 2 et 4*
  - *MPEG1 :*
    - 352 x 288 à 25 images par seconde
    - Inclut le format audio MPEG-1 Layer 3 (MP3).
    - *codage JPEG puis codage des différences entre les images*
    - *Qualité télé standard: 1,5 Méga bit/s*
  - *MPEG2 :*
    - *Télé sur ADSL, Télé haute définition , TNT*
    - *15 à 20 mégabit/s pour télé HD*

Dans l'ordinateur "personnel"

- **Carte mère**

Type de circuits

•Microprocesseurs : calculs généraux

•Mémoires: RAM et ROM

•Digital Signal Processor (DSP) : calculs numériques généraux,

spécialisés : addition / multiplication, matrices

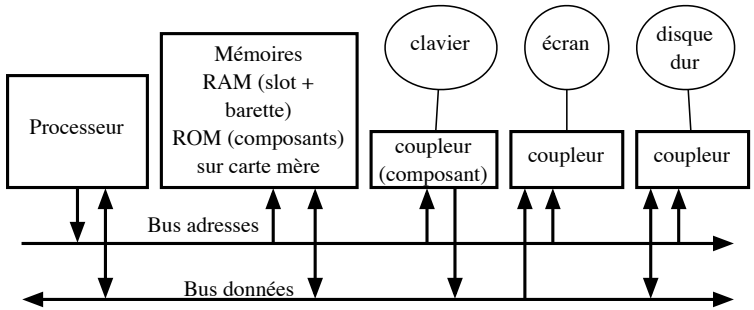
•Accélérateurs divers : compression image / son, cryptographie, ...

•Contrôleurs de périphériques : Ecran, disques, réseau, ...

•Gestionnaires internes : horloges, puissance, ...

Composants de l'ordinateur

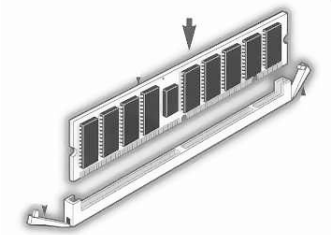
- **Description fonctionnelle**
  - **Bus** : nappe de fils
  - **Processeur** (ou CPU, unité centrale de traitement):
    - coeur de l'ordinateur ; exécute les calculs
  - **Périphériques** :
    - stockage, interfaces avec le monde extérieur
    - vu par le processeur comme des cases mémoires (à travers les coupleurs)



Mémoires

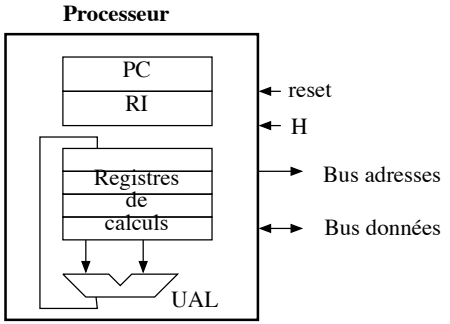
- Mémoire centrale/principale (Vive) : **RAM (Random Acces Memory)**,
  - rapide (quelques **dizaines de nano-secondes**),
  - contient les programmes en cours d'exécution, électrique donc non permanente
- Mémoire morte **ROM (Read only Memory)**:
  - contient les premiers programmes exécutés à la mise sous tension de l'ordinateur
- Mémoire **cache** :
  - encore plus petite et plus rapide (**quelques nano-secondes**),
  - contient aussi une partie des programmes en cours d'exécution, on l'oublie pour l'instant
- Mémoire **secondaire** :
  - disque dur,
  - beaucoup plus lente (**quelques milli-secondes**, facteur 10 puissance 6) mais sauvegarde permanente (magnétique)
  - Bientôt remplacé par la technologie SSD (mémoire flash comme les clés USB)

Mémoires RAM

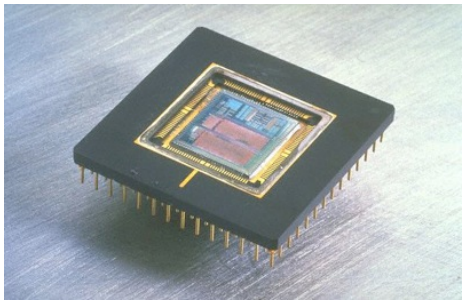


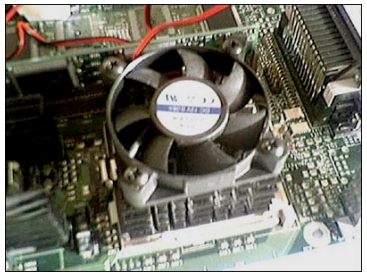
Processeur

- Contient quelques cases de mémoires appelées "registres" plus rapide que la RAM (facteur 10)
- Architecture X bits (nombre de bits des registres)
- Effectue des calculs grâce à une Unité Arithmétique et Logique (**UAL**)
- 1 registre particulier : **PC** (compteur programme) qui contient l'adresse de l'instruction en cours d'exécution
- 1 registre qui contient l'instruction machine en cours d'exécution (**RI**)
- **Reset** pour recommencer au début
- L'Horloge (**H**) cadence le rythme de calcul du processeur



Processeur





Différentes "formes" des programmes

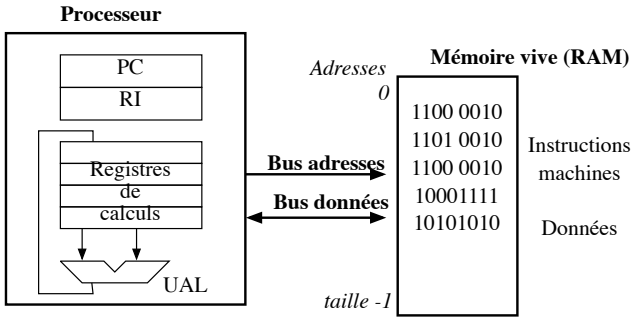
- **Programme** : suite d'**actions** (décidée par un **algorithme**) sur des **données**
- Programme sous différentes formes/langages :
  - Programme **source** dans un langage de haut niveau (ADA, C, PASCAL, JAVA…
  - Programme en **langage machine** (binaire) seul **exécutable** par le matériel
  - **Programme d'assemblage** : langage intermédiaire lisible (par l'homme) des instructions machines
- Passage d'un à l'autre grâce à un programme particulier : le **compilateur**
- Sur les premiers ordinateurs, les compilateurs n'existaient pas, les gens programmaient en binaire à l'aide d'interrupteurs.
- Les trois langages (source, assemblage, machine) sont codés en binaire : codes ASCII ou instructions machines

Compilateurs et exécutables

- **Langage machine propre à un processeur**.
- On ne peut pas utiliser un « exécutable » sur différents processeurs (PC et MAC par exemple)
- Il faut recompiler le programme source
- "Binaire universel" ? : contient les différents exécutables
- **Compilation**
  - Source -> Binaire/exécutable
  - Si plusieurs sources étape particulière : "Editions de liens"
- Interprétation : transformation faite pendant l'exécution du programme.
- Interprétation n'existe quasiment plus (ou après une transformation intermédiaire)
- Exemple d'interpréteur: Machine JAVA: programme qui interprète du "Bytecode Java" (forme intermédiaire entre source et exécutable)

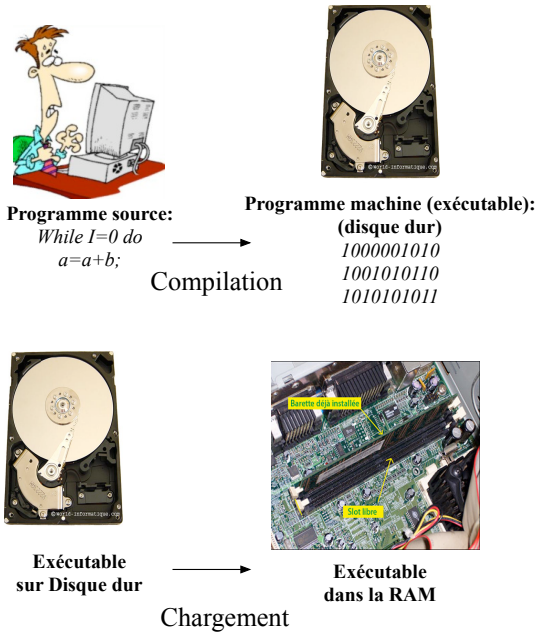
Principe de l'exécution des programmes

- Instructions machine et données sont mises en mémoire vive (RAM)
  - C'est déjà un programme qui fait cela
- Le processeur lit en mémoire successivement les instructions machines en mettant sur le bus adresse l'adresse de l'instruction à exécuter
- Ces instructions peuvent être de différents types : lecture/écriture des données, calculs
- Enchaînement dans le temps séquentiel suivant l'écriture en mémoire, sauf instruction spéciale de branchement



Exécution des programmes

- **Interpréteur de commande:** programme permettant de lancer d'autres programmes (fait partie du système d'exploitation). Lancement d'un programme en donnant le nom de l'exécutable (ou double click sur une icône)
- **Chargeur** : programme particulier qui s'occupe de charger en mémoire l'exécutable lors de son lancement
- **Processus**: Programmes en cours d'exécution, il peut y en avoir plusieurs en même temps : entrelacement dans le temps de leur exécution par le processeur



Logiciels

- Programmes dans l'ordinateur
- Classement suivant leurs fonctionnalités
- Système d'exploitation :
  - Programmes basiques pour gérer les différents composants.
  - Propre à une machine
    - Interpréteur de commande (shell UNIX)
    - Chargeur
    - Dialogue avec les périphériques
    - …
- Compilateur : transformation source -> exécutable
- Applications diverses (éditeurs de texte, navigateur Web...)