

Informatique Générale

Responsable du cours

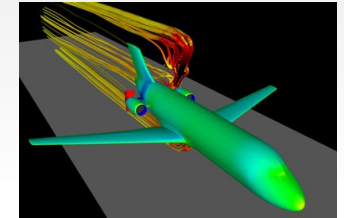
Jacques Farré

Jacques.Farre@unice.fr

<http://deptinfo.unice.fr/~jf/InfoGene>

Pourquoi l'informatique ?

- Les «ordinateurs» sont omniprésents :
 - Dans la vie quotidienne : téléphonie, photographie, guichets automatiques, automobiles et transports, web, jeux ...
 - Dans la vie professionnelle : de l'artisan (logiciels de comptabilité, de facturation ...) à l'ingénieur (logiciels de simulation, de conception assistée ...)
- En comprendre les fondements
 - Pour acquérir une rigueur de pensée (**les ordinateurs n'aiment pas «l'à peu près»**) et une compétence supplémentaire, pour les utiliser plus efficacement, pour être capable de dialoguer avec un informaticien (qui ne fait pas toujours les efforts nécessaires !)
 - Pour ne pas se faire piéger («phishing» par ex.) ou raconter des «bobards» («c'est la faute à l'ordinateur, j'y peux rien moi»)



simulation de propulsion
(ONERA)

Qu'est-ce que l'informatique ?

- INFORMation autoMATIQUE
- Définition de l'Académie Française : **Science** du traitement **rationnel**, notamment par machines **automatiques**, de **l'information** considérée comme le support des connaissances et des communications, dans les domaines technique, économique et social
 - **Sciences et Technologies de l'Information et de la Communication (STIC)**
- Rien à voir avec « bidouiller » son PC, maîtriser PhotoShop, tenir un blog ou connaître les ruses d'un jeu vidéo (même si ça peut y aider parfois !)

« L'informatique n'est pas plus la science des ordinateurs que l'astronomie n'est celle des télescopes. » Edsger Dijkstra



www.camillejourdain.fr/wp-content/uploads/2008/03/humour-blog2.jpg

Qu'est-ce que l'informatique ?

- *Science* : donc des théories et des modèles
- *Traitement rationnel* : c'est à dire l'utilisation de méthodes précises, fondées, répliquables
- *Machines automatiques* : qui seront capables de mettre en œuvre ces méthodes dans un «langage» qu'elles peuvent «comprendre»
- *Information* : texte, image ou son (et mêmes odeurs, saveurs et toucher), représentés sous une forme manipulable par la machine en fonction de sa technologie



Logiciel libre RoseGarden
(www.rosegardenmusic.com)

Disciplines de l'informatique

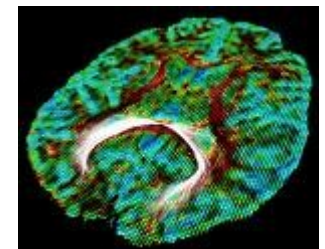
- Informatique théorique (algorithmique, calculabilité, complexité, graphes, langages formels ...)
- Programmation, génie logiciel
- Réseaux, logiciels de télécommunications, sécurité des échanges d'information
- Logiciels de base (système, compilateurs ...)
- Systèmes embarqués, robotique
- Images, son, multimédia, interfaces homme/machine, infographie ...
- Systèmes d'information, ingénierie des connaissances ...
- Calcul scientifique, optimisation, intelligence artificielle, bio-informatique, traitement des langues ...



robot mars explorer (NASA)

Quelques domaines d'application de l'informatique

- Informatique de gestion
 - Compatibilité, facturation, paye, gestion des stocks et du système de production, gestion des relations clients, banques et bourse, aide à la décision ...
- Informatique industrielle et technologique
 - Conception et fabrication assistées, modélisation et simulation de systèmes complexes, informatique embarquée, télécommunications et réseaux ...
- Internet
 - e-commerce, recherche d'informations, sécurité ...
- Et aussi disciplines scientifiques, médicales, sciences humaines et sociales, arts ...



cartographie de
connexions du cerveau
(INRIA)

Les métiers de l'Informatique

- Métiers de l'exploitation
 - Technicien de maintenance, technicien support (Hot Liner), administrateur de système d'information, de système, de réseau, webmestre ...
- Métiers de la conception et du développement
 - Analyste-programmeur, concepteur de logiciel, architecte de systèmes d'information, web designer, ergonomiste ...
- Métiers de la recherche
 - En informatique « pure » ou appliquée à d'autres domaines
- Métiers du conseil et de l'expertise
 - Mise en œuvre de systèmes d'information, protocoles de sécurité, impact sur l'organisation de l'entreprise ...
- Métiers du marketing
 - Ingénieur technico-commercial, chef de produit



Brève histoire de l'informatique des cailloux à la puce

Invention
des chiffres



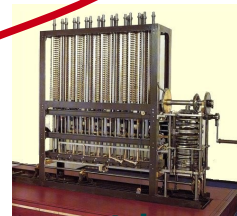
Abaques et bouliers



Machines à calculer
mécaniques



Machine
programmables



Ordinateurs
électroniques



Ordinateurs
portables



-3500

-500

1600

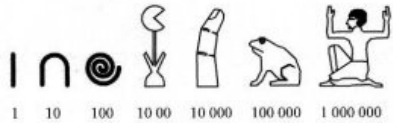
1800

1950

2000

Evolution des concepts

systèmes de numération additifs



(chiffres égyptiens)

systèmes positionnels sans zéro



(9038 en babylonien
 $2 \times 3600 + 30 \times 60 + (30 + 8)$)

Introduction du zéro, base décimale

calculs manuels

	2	3	8	
	2	3	8	1
	6	9	2	3
3	0	9	4	

($238 \times 13 = 3094$)

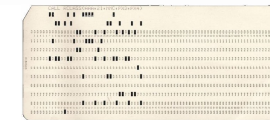
calculs automatisés



(machine de Schikard)

Algèbre de Boole

machines programmables



(carte perforée)

-2000

500

1600

1800

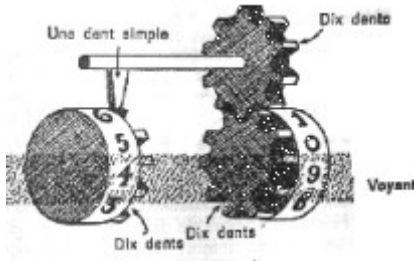
Evolution des techniques



boulrier



calculi sumériens



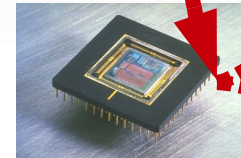
mécanique



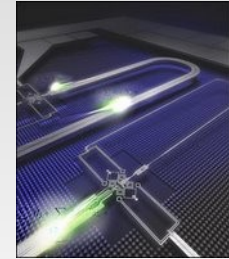
tube à vide



transistor



circuit intégré



quantique ?

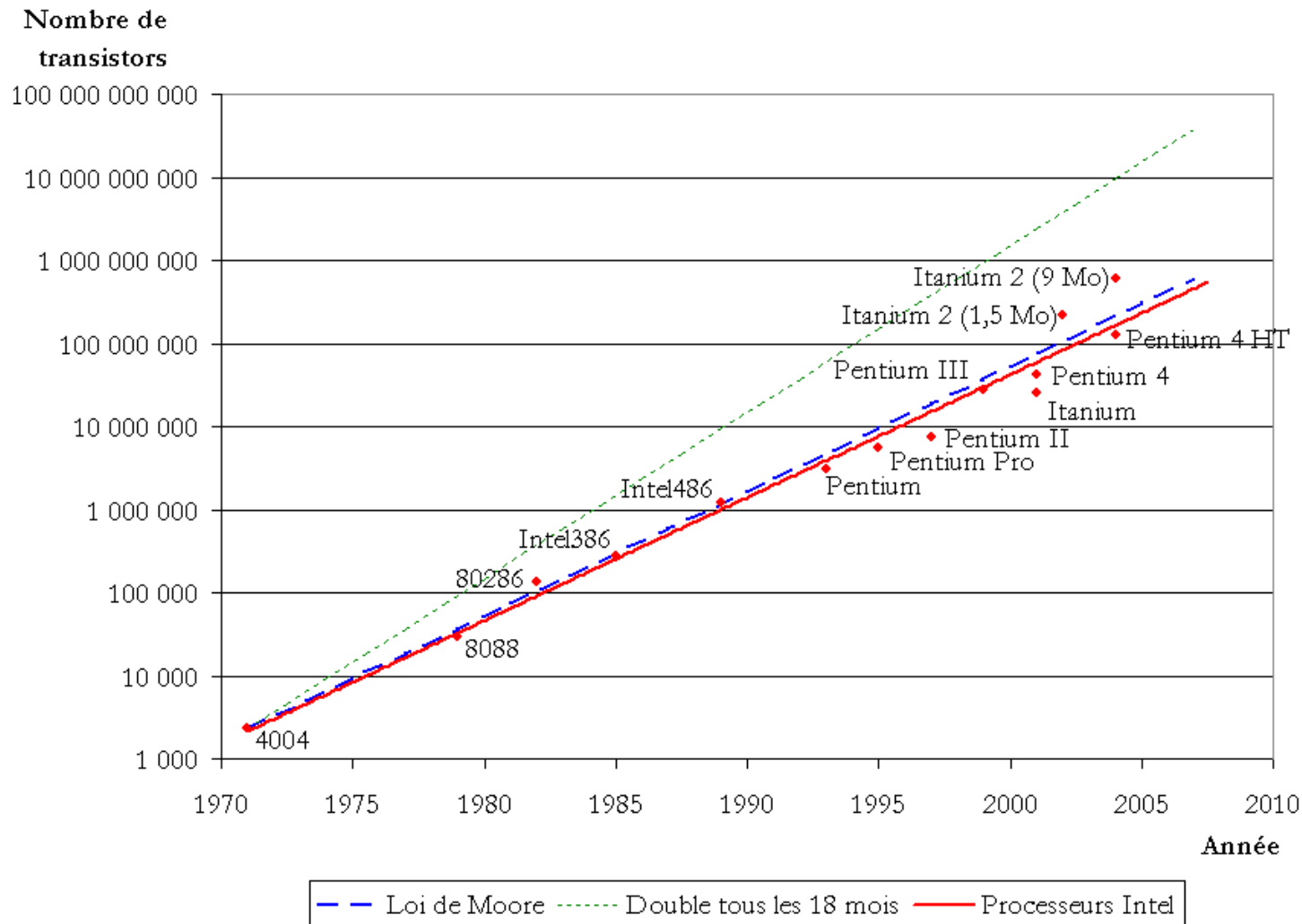


moléculaire ?

manuel

électronique

Intégration sur une puce : jusqu'où ? (exemple avec les processeurs Intel)



- **MIPS (million d'instructions par seconde)**
- **4004 : 0,06**
- **8088 : 0,3**
- **80286 : 1**
- **80486 : 20**
- **Pentium : 100**
- **Pentium 4 : 5000**
- **Core 2 Duo: 20000**
- **Core 2 Quad : 50000**

(source Wikipédia)

Brève histoire de l'informatique

des cailloux à la puce

Préhistoire, antiquité et moyen age

- Au début les hommes comptaient sur leurs doigts et avec des pierres ou des bâtons
- Développement de l'agriculture et du commerce, donc des opérations de calcul
 - besoin de représenter les nombres à l'aide de chiffres (Sumer, -3500)
 - besoin de méthodes et de moyens de calculs : utilisations d'abaques (plateaux sur lesquels on déplace des cailloux ou des jetons) et de bouliers : Moyen-orient, Russie et Chine
- **Définition de la logique par Aristote** (5^e siècle avant JC)
- Invention de la numération décimale de position et du 0 (Inde, 4^e siècle après JC)
 - il faut attendre le 10^e siècle pour qu'ils arrivent en Europe (Espagne via les arabes), et commencent à être largement adoptés au 14^e siècle



La comptabilité d'un chasseur préhistorique



Abaque romain



Chiffes indiens

Brève histoire de l'informatique des cailloux à la puce

Plus tard (17^e siècle essentiellement)

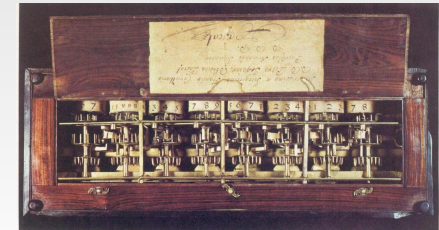
- Inventions des logarithmes (Neper) : permet de ramener multiplication et division à somme et différence
- Codage des lettres de l'alphabet par Bacon (1623, *alphabet bilitère*, a=AAAAA, b=AAAAB, C=AAABA, D=AAABB, E=AABAA... z=BABBB) pour coder des messages secrets :
 - question : pourquoi faut-il 5 lettres pour en coder une ?
 - texte en clair et codage bilitère (A si lettre italique, B sinon)
N e p a r t e z s u r t o u t p a s s a n s m o i
A A B A B B A A B B B A B B A A A B A A B A B B B
 - Texte codé : *fuyez*
- Le code bilitère ouvre la voie à l'arithmétique binaire (Leibnitz, 1703, qui la relie à un symbole chinois du 3^e siècle avant JC), puis à **l'algèbre de Boole** (1854), base théorique du fonctionnement des ordinateurs



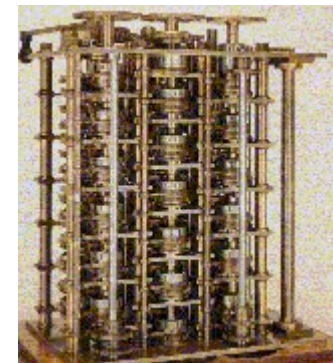
Brève histoire de l'informatique des cailloux à la puce

Les premières machines à calculer

- Invention par Pascal (agé de 19 ans !) de la *Pascaline* (1642) : additionne et soustrait par un système de roues dentées ; il en existe plusieurs exemplaires dans les musées
 - Inspirée de la machine de Wilhelm Schickard (1623)
 - Perfectionnée par Leibnitz (1694) en permettant multiplications et divisions (par additions ou soustractions successives)
- Invention par Jacquard (1805) de la notion de «programme» : cartes perforées pour commander des métiers à tisser
- Les machines de Babbage
 - *machine à différences* (1822), jamais terminée mais reconstruite en 1991 (25000 pièces, 4,5 tonnes)
 - *machine analytique* (1843) : on y retrouve les composants d'un ordinateur : processeur, mémoire, saisie et affichage des données ; programmée par Ada Lovelace



La Pascaline



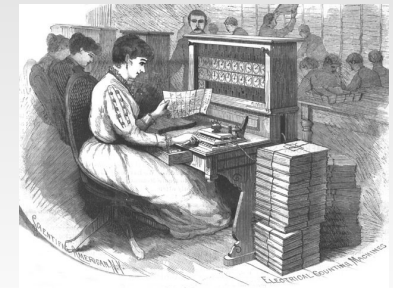
La machine à différences

Brève histoire de l'informatique

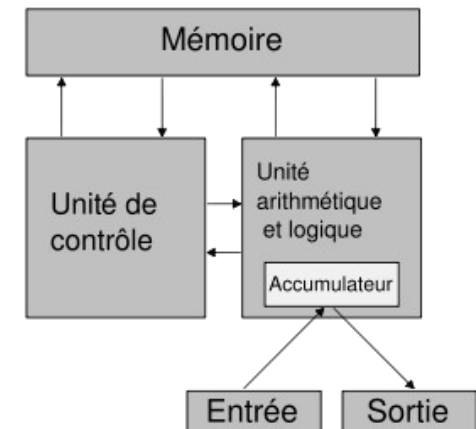
des cailloux à la puce

Les calculateurs électro-mécaniques et électroniques

- Hollerith dépose un brevet pour une machine à calculer automatique (1884), et fonde (1896) la société qui deviendra IBM
- Construction au MIT d'un calculateur analogique (1925)
- Turing propose sa définition de machine (1936), outil capital pour l'informatique théorique
- L'ENIAC est construit en 1946 en partie sur les principes de von Neumann : il pèse 30 tonnes, occupe 72 m², est équipé de 19 000 lampes. Il calcule en décimal, repose sur des tubes à vide, et n'a pas de programmes enregistrés. C'est encore un calculateur, pas un ordinateur universel
- Von Neumann définit l'architecture d'un **ordinateur** universel (EDVAC, 1949)



Architecture de von Neumann



Brève histoire de l'informatique

des cailloux à la puce

Les premiers ordinateurs

- Création de la théorie de l'information (Shannon, 1948) : mesure d'une quantité d'informations en chiffres binaires (bits)
- Construction du *Manchester Mark I*, premier ordinateur (1948) à programme enregistré
- Univac, premier ordinateur commercialisé (1951) par Remington Rand (15 exemplaires vendus) ; premier ordinateur français (pour l'armée)
- Premier ordinateur construit en série par IBM (1953) : 900kg, vendu \$500 000 à plus de 1000 exemplaires
- Invention du mot *ordinateur* (1955), initialement par IBM
- Apparition des ordinateurs à transistors (1956), et création du premier disque dur (IBM : 1000Kg, 5 Mega)
- Invention du terme *informatique* (1962)



IBM 650

Brève histoire de l'informatique des cailloux à la puce

Plus récemment

- Premier mini-ordinateur (Digital Equipment Corporation, 1963), et premier super-ordinateur (Control Data Corporation, 1964)
- Premiers ordinateurs à circuits intégrés (Burroughs, 1968)
- Création du système d'exploitation Unix (1969), qui a inspiré Linux
- Création du réseau Arpanet, ancêtre d'Internet (1969, 4 ordinateurs)
- Apparition des mémoires en circuits intégrés (1970)
- Premier courrier électronique (1971)
- Premier micro-processeur (Intel, 1971), puis premier micro-ordinateur par la société française R2E (1973)
- Naissance de Microsoft (1975) et d'Apple (1976)
- Création de l'Internet Protocol (1982)
- Premier logiciel libre (GNU, Richard Stallman, 1983), puis fondation de la Free Software Foundation (1985)



PDP 8 de DEC



Arpanet en 1971



Micral N de R2E



Brève histoire de l'informatique des cailloux à la puce

L'informatique moderne

- Apparition des interfaces graphiques et de la souris (1984, Macintosh d'Apple, X Window au MIT pour Unix) : l'ordinateur tel que vous le connaissez est né !
- Commercialisation de Windows (1985, pas vraiment graphique), il faudra attendre Windows 95, avec Internet Explorer en option)
- Premier virus (transmis par disquette, 1986) et premiers anti-virus (gratuits, 1988), premier ver transmis par Internet (1988)
- Création de Linux (Linus Torvalds, 1991), système d'exploitation libre
- Création du Web (Tim Berners-Lee, 1991) et premier navigateur (Mosaic, 1993); création de Yahoo (1994), naissance du W3C (1994)
- Un ordinateur bat un champion des échecs, Kasparov (1994)
- Création de Google (1998)
- Années 2000 : web 2.0, premiers systèmes collaboratifs, de partage et d'échanges (wiki, forums, peer to peer ...), « cloud computing »



Macintosh d'Apple



Deep Blue d'IBM, super-ordinateur massivement parallèle

Les logiciels libres

(d'après www.gnu.org/philosophy/free-sw.fr.html)

- L'expression « Logiciel libre » fait référence à la liberté et pas au prix. Pour comprendre le concept, vous devez penser à la « liberté d'expression », pas à « l'entrée libre »
 - C'est la liberté pour les utilisateurs d'exécuter, de copier, de distribuer, d'étudier, de modifier et d'améliorer le logiciel. Plus précisément, elle fait référence à quatre types de liberté pour l'utilisateur du logiciel :
 - La liberté d'exécuter le programme, pour tous les usages (liberté 0)
 - La liberté d'étudier le fonctionnement du programme, et de l'adapter à vos besoins (liberté 1). Pour ceci l'accès au code source est une condition requise
 - La liberté de redistribuer des copies, donc d'aider votre voisin, (liberté 2)
 - La liberté d'améliorer le programme et de publier vos améliorations, pour en faire profiter toute la communauté (liberté 3). Pour ceci l'accès au code source est une condition requise
- En France, AFUL (www.iful.org), APRIL (www.april.org), et d'autres



Les logiciels libres

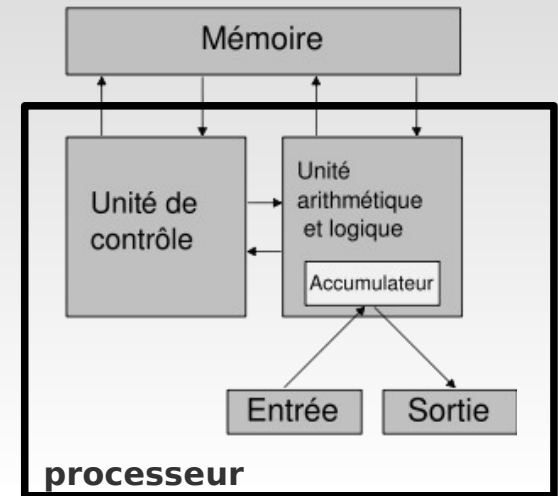
(d'après fr.wikipedia.org/wiki/Portail:Logiciels_libres)

- On trouve des logiciels libres (et le plus souvent gratuits) pour de nombreuses applications
 - Tout d'abord les systèmes d'exploitation (Linux notamment)
 - Suites bureautique (par exemple OpenOffice)
 - Navigateurs Internet (par exemple Firefox) et email (Thunderbird entre autres)
 - Création de sites Web
 - Pare-feu, antivirus
 - Lecteurs audio, vidéo et multimédia, création graphique ...
 - Et bien d'autres encore ...
- Ils sont en général disponible pour Linux et pour Windows, et assez faciles à installer
- Industrie des logiciels libres : environ 40000 emplois en France, 1 milliard d'euros de chiffre d'affaire pour 2008



Mais comment ça marche ?

- Reprenons l'architecture de von Neuman
 - Le **processeur** est constitué de circuits électroniques programmables par une suite d'instructions (le programme), en fait une suite de 0 et de 1, indiquant s'il doit passer ou non du courant électrique
 - Les **instructions** sont dans la **mémoire** (en fait elles sont chargées dans la mémoire à partir d'un support externe : disque dur ...)
 - Les instructions peuvent lire ou écrire des données dans des **registres** (*accumulateur* sur le dessin) ou dans des « cases » de la mémoire identifiées par leur **adresse** (un nombre entier)
- Pour un être humain, écrire des suites de 0 et de 1, ce n'est pas très amusant, pas bien lisible, et cause de nombreuses erreurs
 - On a donc inventé les **langages de programmation**, qui étaient d'abord une simple forme symbolique des instructions, puis sont devenus de plus en plus évolués et éloignés de la machine



Les langages de programmation

- Les langages-machine binaire

- Ce sont ceux qui sont stockés sur vos disques (les **.exe** par exemple), et amenés en mémoire par un *chargeur*

```
00101110011...
```

- Les langages-machine symboliques (ou langage d'assemblage, langage assembleur)

- Permettent à un être humain d'écrire des instructions de base sous forme symbolique (action de l'instruction et adresses en mémoire) ; ils sont traduits en binaire par un programme appelé *assembleur*

```
load AX, I  
add AX, 1  
store AX, I
```

- Les langages de haut niveau

- Permettent de s'affranchir des instructions de la machine et de programmer dans un langage plus proche des algorithmes

```
I := I + 1
```

Les langages de haut niveau

- Les premiers langages étaient *impératifs*, par exemple Fortran (1958, encore utilisé de nos jours), mais de nouveaux paradigmes (modèles) de programmation ont vu le jour depuis
 - Paradigme impératif : reste assez proche du modèle de von Neumann
 - Représentants les plus connus : Fortran, Basic, Algol, C, Pascal, Ada
 - Paradigme fonctionnel : basé sur le λ -calcul (Church, 1930), repose sur la notion d'application de fonction
 - Représentants les plus connus : Lisp, Scheme, ML, CaML, Haskell
 - Paradigme logique : définit des faits et des règles logiques permettant d'inférer d'autres faits
 - Représentant le plus connu : Prolog
 - Paradigme objet : sous-tend l'idée qu'un programme est un assemblage de composants logiciels
 - Représentants les plus connus : Smalltalk, C++, Java, C#, Python

```
s := 0;
for i := 1 to n do
  s := s + i
```

```
let somme n =
  if n = 0 then 0
  else n + somme n-1
```

```
somme(0,0)
somme(n,s) :-
  m is n-1, somme(m,r),
  r is s + n
```

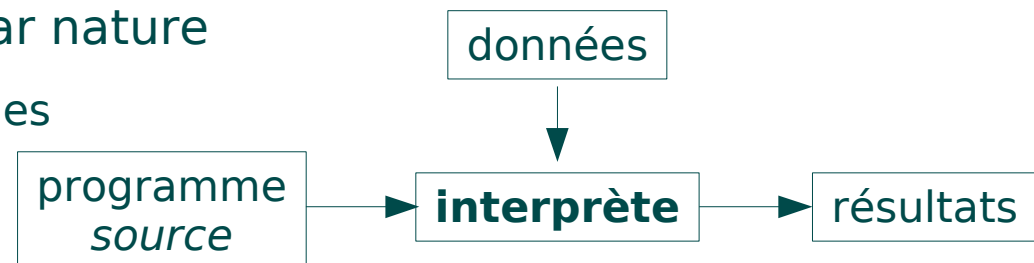
Comment la machine « comprend » un langage de programmation ?

- Il faut un programme qui
 - soit traduit le texte de votre programme en code binaire, on a affaire à un **compilateur** : le programme exécutable qui en résulte est rapide (si vous avez bien programmé, évidemment)



- Concerne surtout les langages impératifs
- ou exécute votre programme (gardé en mémoire sous une forme plus appropriée que la forme textuelle), on a alors un **interpréteur** (ou **interprète**) : moins rapide par nature

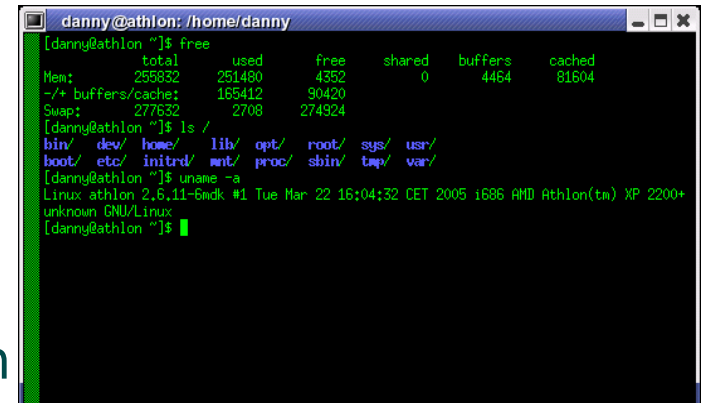
- Concerne les langages logiques ou fonctionnels, qui peuvent être néanmoins compilés



- Modèle mixte : le compilateur traduit dans le langage d'une *machine virtuelle*, interprété par un simulateur de cette machine (Java par exemple)

Systeme d'exploitation

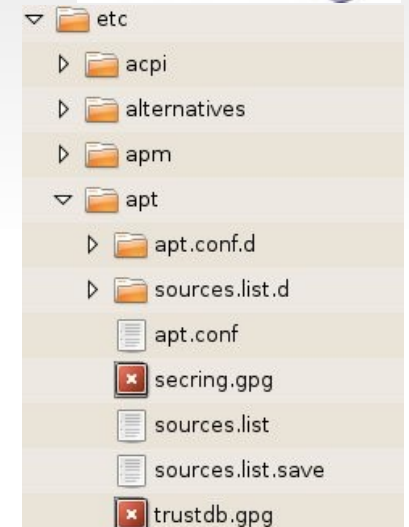
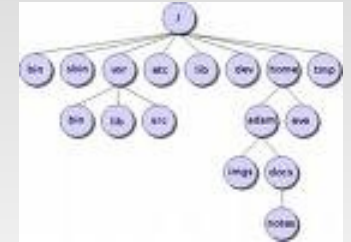
- C'est le programme responsable d'organiser les échanges entre les ressources matérielles de l'ordinateur et les programmes lancés par l'utilisateur (ou activés derrière son dos) : Linux, MacOS, Windows ...
 - Un **noyau** pour :
 - Gestion des périphériques (pilotes – *drivers*) et du réseau
 - Gestion de la mémoire et ordonnancement des différents processus (les programmes qui tournent)
 - Système de fichiers
 - Une **interface utilisateur** (lignes de commandes – clavier ou interface graphique – clavier – souris - fenêtres)
 - Des **bibliothèques** (sous-programmes souvent utilisés) et des outils de gestion et d'observation du système



```
danny@athlon: /home/danny
[danny@athlon ~]$ free
              total        used        free      shared    buffers     cached
Mem:           255832      251480         4352           0          4464       81604
-/+ buffers/cache: 165412      90420
Swap:          277632         2708       274924
[danny@athlon ~]$ ls /
bin/  dev/  home/  lib/  opt/  root/  sys/  usr/
boot/ etc/  initrd  mnt/  proc/  sbin/  tmp/  var/
[danny@athlon ~]$ uname -a
Linux athlon 2.6.11-6mdk #1 Tue Mar 22 16:04:32 CET 2005 i686 AMD Athlon(tm) XP 2200+
unknown GNU/Linux
[danny@athlon ~]$
```

Systeme de gestion de fichiers (SGF)

- SGF : structure de données *arborescente* stockant les informations des mémoires secondaires (disque dur, CD-ROM, clé USB, ...)
Répertoires (dossiers) = nœuds de l'arbre, fichiers = feuilles
- Il assure plusieurs fonctions :
 - Créer/détruire/modifier un fichier
 - Allocation de la place sur mémoires secondaires
 - Localisation des fichiers (*chemin d'accès*)
 - Sécurité et contrôle (propriétaire, qui peut lire/écrire ...)
- Le journal est la partie du SGF qui note les opérations d'écriture tant qu'elles ne sont pas terminées afin de garantir l'intégrité des données en cas d'arrêt brutal
- Principaux systèmes
 - FAT32, VFAT (non journalisé, Windows)
 - ext[2,3,4] (Linux), NTFS (Windows), journalisés



Quelques références bibliographiques

- Philippe Breton : *Une histoire de l'informatique*, collection «Point Sciences», Le Seuil, 1990 (une histoire qui vous paraîtra un peu ancienne)
- Solange Ghernaouti-Hélie, Arnaud Dufour : *De l'ordinateur à la société de l'information*, collection «Que sais-je ?», 1999
- Jean Marguin : *Histoire des instruments et machines à calculer*, Hermann, 1994
- Christian Piguet et Heinz Hügli : *Du zéro à l'ordinateur. Une brève histoire du calcul*, Presses Polytechniques Romandes, 2004
- Jean-Luc Charbert et al : *Histoire d'algorithmes*, Belin, 1993
- Amy Dahan-Dalmeco et Jeanne Peiffer : *Routes et dédales. Histoire des mathématiques*, collection «Point Sciences», Le Seuil, 1986
- Georges Ifrah : *L'histoire universelle des chiffres*, collection «Bouquins», Robert Laffont, 1994
- Dominique Nora, Roberto Di Cosmo : *Le Hold-up Planétaire*, Calmann-Lévy, 1998 (contre Microsoft)

Quelques références sur le web

- Histoire des maths :
<http://www.dma.ens.fr/culturemath>
<http://www.ilemaths.net/encyclopedie/index.html>
<http://www.roosoft.com/mathegyptiennes>
<http://pagesperso-orange.fr/therese.eveilleau>
- Histoire de l'informatique
<http://www.histoire-informatique.org>
<http://www-ipst.u-strasbg.fr/pat/internet/histinfo>
<http://www.commentcamarche.net/histoire/ordinateur.php3>
- Généralités sur l'informatique
<http://fr.wikipedia.org/wiki/Informatique>
http://en.wikiversity.org/wiki/Introduction_to_Computer_Science (la version française est en retard)
<http://zabaque.uqac.ca>
- La recherche en informatique : <http://interstices.info>

Ce que traitera la suite du cours (planning prévisionnel)

- Algèbre de Boole (J. Farré)
- Représentations des nombres (J. Farré)
- Histoire d'algorithmes et de machines (J. Farré)
- Composants matériels d'un ordinateur (F. Huet)
- Systèmes d'exploitation et réseaux (F. Huet)
- L'internet et le web (Ph. Renevier)