



# Histoire de l'ordinateur

---

Tanmoy Mondal

[tanmoy.mondal@lirmm.fr](mailto:tanmoy.mondal@lirmm.fr)



# Objectifs de ce cours



1. Connaître les grandes étapes de l'histoire de l'informatique ;
2. Découvrir l'architecture de Von Neumann.

# Besoin de calculer

- Tablettes scolaires mathématiques de l'époque **paléo-babylonienne** (**deuxième millénaire** avant notre ère) ;
- L'ordinateur est né du besoin de calculer de manière :
  - toujours plus complexe ;
  - toujours rapide.
- ⇒ Automatiser le calcul.



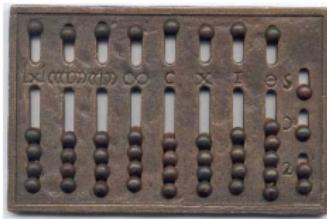
# L'histoire du Ordinateur- Tally Stick

- Un "Tally Stick" était un ancien dispositif de mémoire
- Utilisé pour **enregistrer et documenter** des **numéros, des quantités** ou même des **messages**
- Marco Polo (1254-1324), qui mentionne l'utilisation du décompte en Chine
- Il était utilisé pour :
  - La messagerie et la planification
  - En particulier dans les transactions financières et juridiques



# L'histoire du Ordinateur – Abaque (Abacus)

- Un abaque ("Abacus") est un dispositif mécanique utilisé pour aider un individu à **effectuer des calculs mathématiques**
- L'abaque a été inventé à **Babylone en 2400 Av. J.-C.**
- Le abaque, que nous connaissons le mieux a été utilisé pour **la première fois en Chine vers 500 Av. J.-C.**
- Il effectuait des opérations arithmétiques
- Les abaque sont encore **d'usage courant dans certains pays**
- Les marchands, les **commerçants et les employés dans certaines régions d'Europe orientale, en Russie, en Chine et en Afrique** utilisent des abaque
- Ils sont toujours utilisés pour **enseigner l'arithmétique aux enfants**
- Certaines personnes qui **ne peuvent pas utiliser une calculatrice**, peuvent utiliser un abaque



# Fondations – (Napier's Bones)

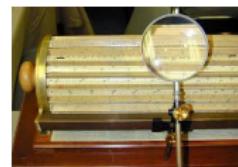
- *John Neper* (1614) : Permettez à l'opérateur de se multiplier, divisez et calculez les racines carrées et cubiques
- En déplaçant les barres et en les plaçant dans des planches spécialement construites



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	
2	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	
3	3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	
4	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100	
5	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	
6	6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126	132	138	144	150	
7	7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147	154	161	168	175	
8	8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	
9	9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	

# Fondations – (Slide Rule)

- *William Oughtred* (1622) : Est basé sur les idées de Napier sur les logarithmes
- Utilisé principalement pour :
  - Multiplication
  - Division
  - Les Racines
  - Logarithmes
  - Trigonométrie
- Non utilisé normalement pour la soustraction et l'addition



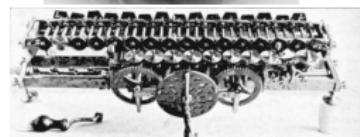
## Fondations – (Pascaline)

- Blaise Pascal (1642) : Première machine à calculer, la pascaline (principe de roues dentées)
- Cette machine pouvait additionner et soustraire des nombres de six chiffres et prenait en compte les retenues !
- C'était trop cher



# Fondations – (Stepped Reckoner)

- Gottfried Leibniz (1673), mathématicien :
- Améliore la machine de Pascal en y ajoutant un mécanisme
- Permettant d'automatiser l'**exécution répétitive d'additions et de soustraction**
- La première machine à calculer autorisant les 4 opérations arithmétiques était née
  - Additions
  - Soustraction
  - Multiplication
  - Division



# Automatisation du Tissage – (Basile Bouchon)

- Basile Bouchon était un pionnier français pour automatiser les modèles du tissage
- En 1725, il inventa un moyen de contrôler **un métier à tisser** avec un **ruban de papier perforé**
- Il automatise **partiellement le processus de configuration** du tirant
  - Dans le métier à cordon, un opérateur soulève les fils de chaîne à en utilisant de cordes
- Ce était la première application industrielle d'une machine semi-automatisée



# Automatisation du Tissage – (Jean-Baptiste Falcon et Jacques de Vaucanson)

## Jean-Baptiste Falcon :-

- Un assistant du Basile Bouchon
- En utilisant des cartes rectangulaires qui étaient assemblées en une boucle (1728)
- Il a augmenté le **nombre de cordes** pouvant être manipulées en **organisant les trous** en rangées

## Jacques de Vaucanson :-

- Cela élimina les erreurs de levage des fils
- Mais il fallut encore **un opérateur supplémentaire pour le contrôler**
- **La première tentative d'automatisation** fut faite par Jacques de Vaucanson en 1745.

## Fondations – (Jacquard Loom)

- Le métier à tisser Jacquard est un métier à tisser mécanique, inventé par **Joseph-Marie Jacquard** en 1804
- C'est un métier à tisser automatique contrôlé par des cartes perforées



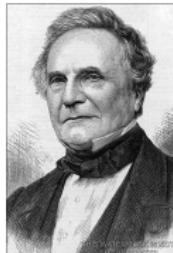
## Fondations – (Arithmometer)

- Une calculatrice mécanique inventée par Thomas de Colmarin 1820
- La première machine à calculer fiable, utile et commercialement réussie
- La machine peut exécuter les quatre fonctions mathématiques de base
- Le premier calculateur fabriqué en masse

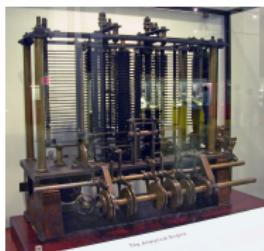


# Fondations – (Moteur Différence et Moteur Analytique)

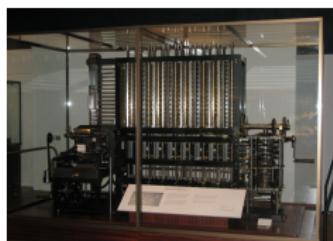
- Considéré comme le père de l'ordinateur (**rapprochement entre commande externe et machine calculer**)
- C'est une calculatrice automatique, mécanique conçue pour **tabuler les fonctions polynomiales**
- Inventé par **Charles Babbage en 1822 et 1834**
- C'est le **premier ordinateur mécanique**



Moteur analytique

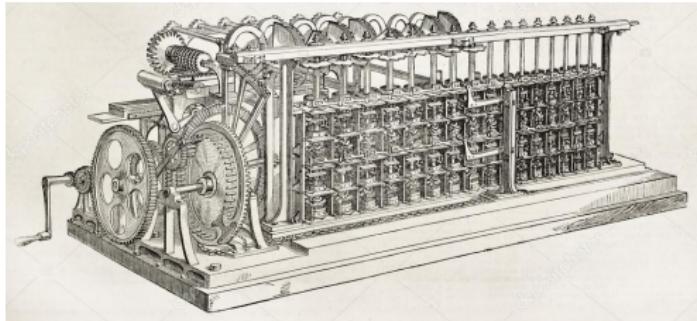


Moteur de Différence



## Fondations – (Calcul de Scheutzian)

- Inventé par Per **Georg Scheutz** en 1843.
- Basé sur le moteur de différence de Charles Babbage
- Le **premier calculateur d'impression**



# Fondations – (Premier programmeur informatique)

- En 1840, Augusta Ada Byrons suggère à Babbage qu'il utilise le système binaire
- Elle écrit des programmes pour le moteur d'analyse



```
1 /* This file contains the main C code for the Hello World example. It is part of
2  * the standard library, it provides the functionality to read input and output functionality
3  * to the program.
4 */
5 #include <stdio.h>
6
7 /*
8  * A function (method) declaration. It takes no arguments and outputs "Hello, world" to
9  * standard output when invoked.
10 */
11 void sayHello() {
12     // printf() in C prints the given text (with optional
13     // formatting codes) to standard output.
14     printf("Hello, world!");
15 }
16
17 /*
18 * This is a "main function". It is the entry point of the program which will run the code
19 * defined here.
20 */
21 void main() {
22     // Invoke the sayHello() function.
23     sayHello();
24 }
```

## Fondations – (Machine à tabuler)

- Inventé par **Herman Hollerithin 1890**
- Pour aider à résumer l'information et la comptabilité
- Calculateur à statistiques (tabulatrice)
- Cartes perforées : premiers supports d'entrée-sortie et premières mémoires de masse (mécanographie)



## Fondations – (George Boole)

- George Boole était un mathématicien anglais
- Inventé *Système de logique symbolique* en 1854
- Calcul booléen : fonctions logiques décrivant le fonctionnement d'un système le plus simple possible.



# Les grandes étapes

- *Basile Bouchon (1725)* : Inventé un moyen de contrôler un métier à tisser  
⇒ *Jean-Baptiste Falcon (1728)* : En utilisant des cartes rectangulaires qui étaient assemblées en une boucle sans fin  
⇒ *Jacques de Vaucanson (1745)* : La première tentative d'automatisation fut faite
- *Joseph Jacquard (1805)* : Un métier à tisser automatique contrôlé par des cartes perforées

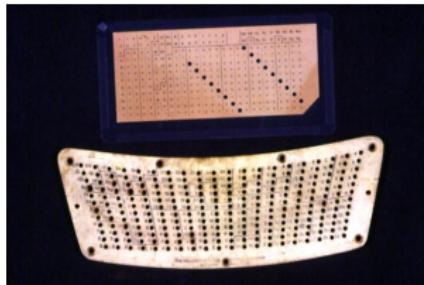


Figure – Cartes perforées

## Les grandes étapes

- *Charles Babbage* (1833) : considéré comme le père de l'ordinateur (rapprochement entre commande externe et machine calculer).  
⇒ Réalisation de sa machine analytique avec l'aide d'*Ada Augusta*, l'ancêtre des ordinateurs.
- *George Boole* (1854) : système de logique symbolique.  
⇒ Calcul booléen : fonctions logiques décrivant le fonctionnement d'un système le plus simple possible.
- *Herman Hollerith* (1890) : calculateur à statistiques (tabulatrice).  
⇒ Cartes perforées : premiers supports d'entrée-sortie et premières mémoires de masse (mécanographie).

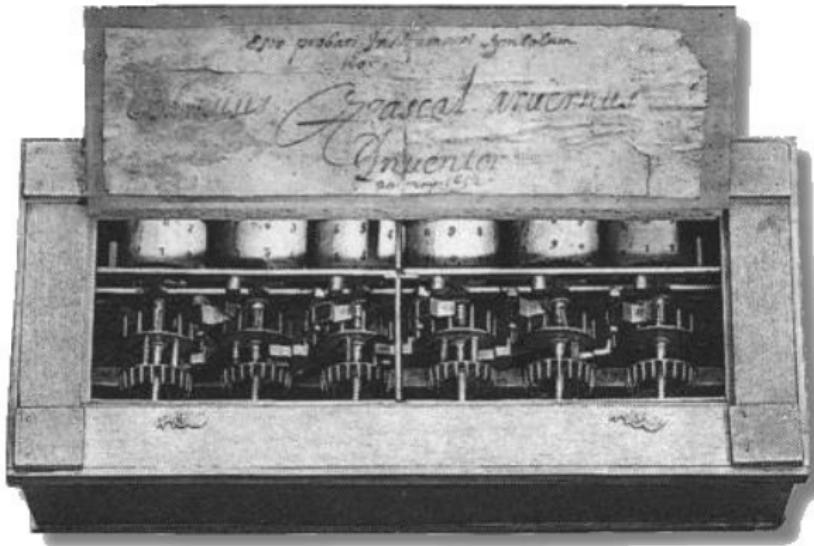


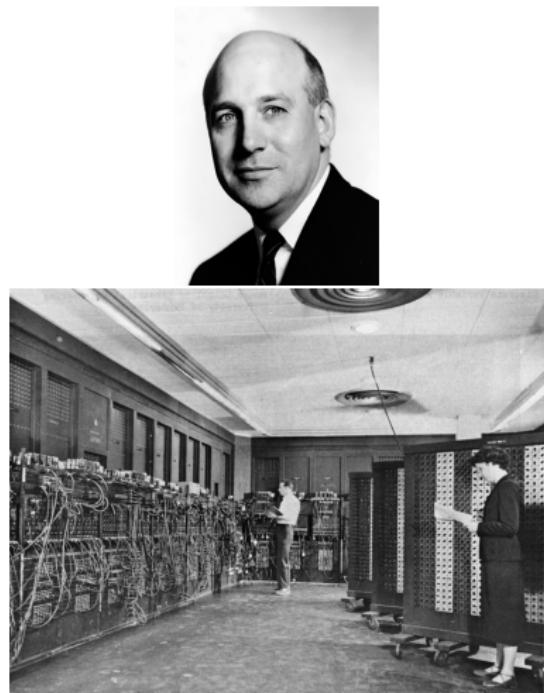
Figure – La pascaline (exposée au CNAM à Paris)

## Un peu d'histoire

- XIème siècle : adoption des chiffres arabes ;
- XVIIème siècle et avant : arithmétique, mathématique ;
- XIXème siècle : les calculateurs ;
- XXème siècle : théorie de l'information + machine universelle ;
- 1945 : architecture de Von Neumann et naissance de l'ordinateur ;
- Depuis 1945 : 4 grandes générations d'ordinateurs se sont succédées.

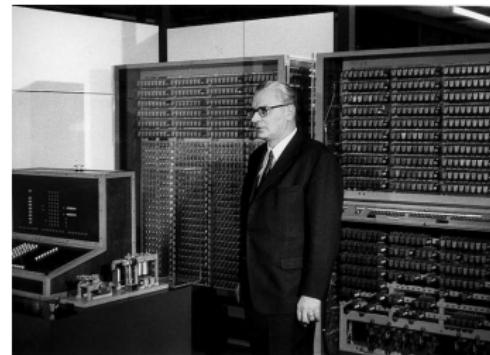
# Naissance de l'ordinateur (Havard Mark 1)

- Également connu sous le nom de "*Automatic Sequence Controlled Calculator (ASCC)*" du IBM
- Inventé par **Howard H. Aiken** en 1943
- Le premier ordinateur électro-mécanique



## Naissance de l'ordinateur (Z1)

- Le premier ordinateur programmable
  - Créé par Konrad Zusein Allemagne de 1936 à 1938
  - L'utilisateur devait insérer une bande-perforée dans un lecteur de bande-perforée
  - Toutes les sorties étaient également générées par la bande-perforée



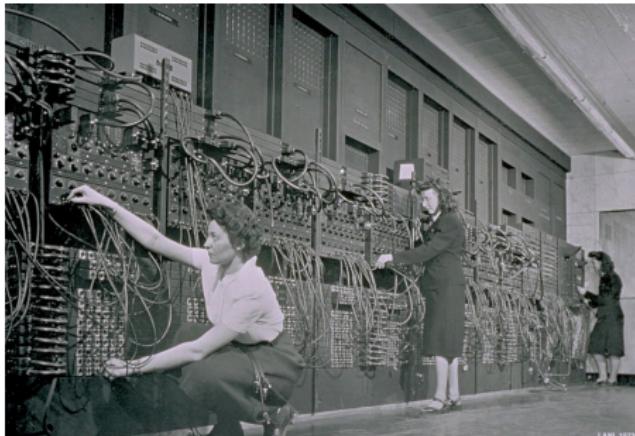
# Naissance de l'ordinateur (Atanasoff-Berry Computer(ABC) )

- C'était le premier ordinateur numérique et électronique
- Inventé par le professeur **John Atanasoff** son étudiant **Clifford Berry** à **Iowa State University** entre 1939 et 1942.



# Construction de l'ENIAC (1946)

- Electronic Numerical Integrator Analyser and Calculator (ENIAC)
- Technologie des tubes à vide (18000, 30 tonnes) ;
- Construit à l'Université de Pennsylvanie par **John Presper Eckert et John W. Mauchly**
- Construit pour développer le modèle Turing-complet
- Multiplication de 2 nombres de 10 chiffres en 3 ms !



## Construction de l'UNIVAC 1 (1946)

- La UNIVAC I (UNIVersal Automatic Computer 1) était le premier ordinateur commercial
- Conçu par **J. Presper Eckert et John Mauchly**
- Prix : 1,25 M \$ - 1,5 M \$
- Unités produites : 46



# Bug informatique (1947)

- Les opérateurs travaillant avec Grace Murray Hopper sur l'ordinateur **Mark II** au **Université Harvard**
- Découvrent un «bogue», **un papillon** logé dans les composants
- Les opérateurs ont apposé le papillon sur le journal-de-l'ordinateur, avec la mention : "**First actual case of bug being found**"

9/9  
0800 Auton started  
1000 stopped - auton ✓ { 1.2700 9.037 897 025  
13'000 020 MP-MC 1.2700 9.037 896 025 connect  
028 PRO ~ 2.130970495  
convk 2.130676495  
Relays 012 ~ 023 fault speech guard test  
in relays more test.  
Relays changed  
1100 Started Cosine Tapg (Sine check)  
1525 Started Multi Adder Test.

1545 Relay #70 Panel F  
moth in relay.

1650 First actual case of bug being found.  
1700 Auton started, closed down.



# Transistor (1947)

- **Bell Labs** développe le transistor (à droite)
- Un commutateur électronique constitué d'un petit morceau de silicium avec impuretés ajoutées
- Il est plus petit
- Consomme moins d'énergie
- Est plus fiable et plus économique à produire que « les tubes à vide »



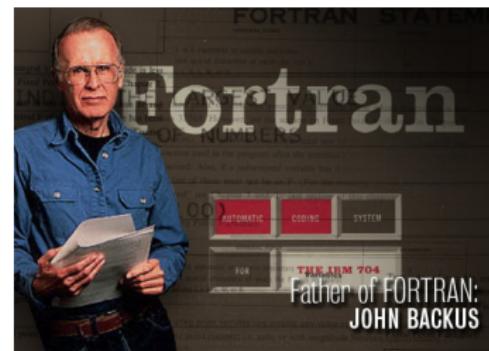
Figure – Les tubes à vide



Figure – Le transistor

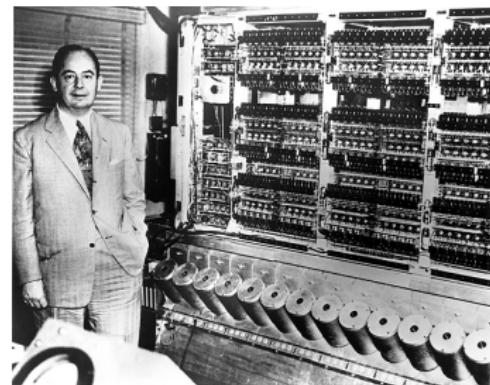
# FORTRAN (1954)

- John Backus et IBM développent **FORTRAN**, le premier langage de programmation et compilateur de haut niveau à succès
- Développé pour des **problèmes scientifiques** et encore largement utilisé aujourd'hui



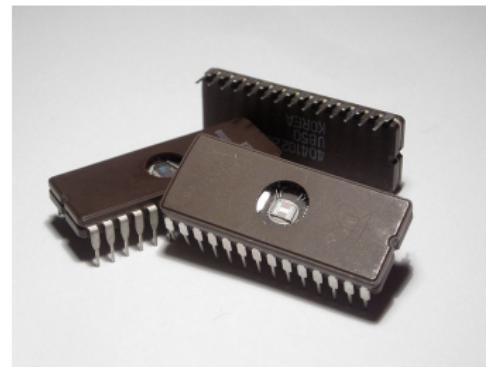
# Construction de l'EDVAC (1952)

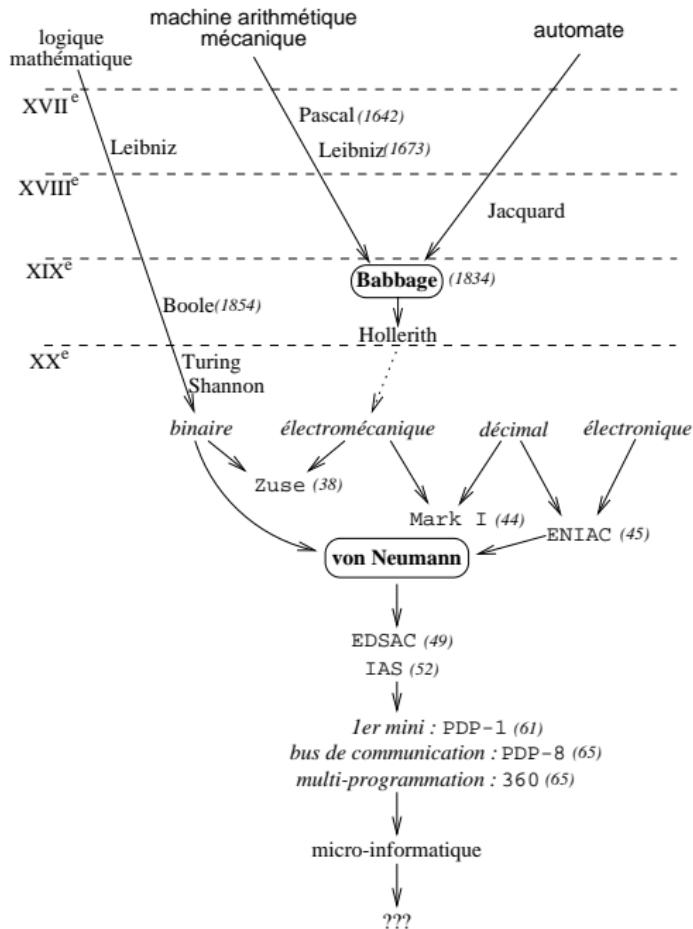
- EDVAC signifie **Electronic Discrete Variable Automatic Computer**
- Le premier ordinateur à programme enregistré
- Conçu par **Von Neumann** en 1952
- Il a une **mémoire pour stocker un programme et aussi les données**



# Circuit Intégré (1958)

- Les circuits intégrés (puces/chips) co-inventés indépendamment par :
  - Jack Kilby de [Texas Instruments](#)
  - Robert Noyce de [Fairchild Semiconductor](#)
- **Transistors, et autres composants électroniques** tous fabriqués sur une seule puce de silicium

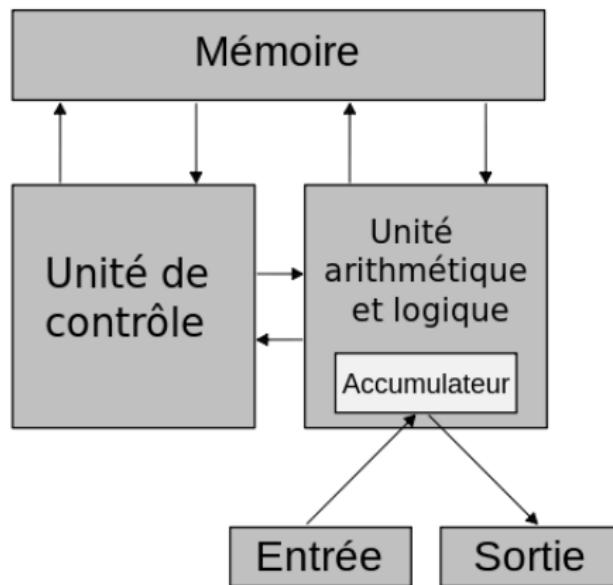




## Principes de l'ordinateur selon Von Neumann

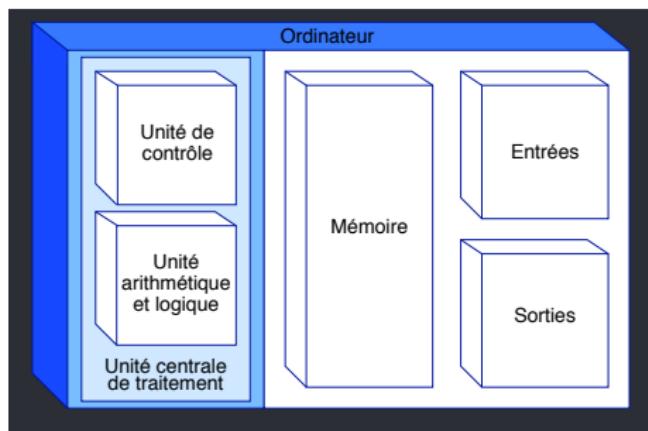
- Machine universelle contrôlée par programme ;
- Programme = séquence d'instructions décrivant comment effectuer une tâche ;
- Instructions du programme codées sous forme numérique binaire et enregistrées en mémoire (tout comme les données utilisées par le programme) ;
- Existence d'instructions permettant les ruptures de séquences.

# Architecture de Von Neumann



# Composants classiques d'un ordinateur

- La mémoire centrale qui contient les données et les programmes à exécuter ;
- L'unité centrale de traitement (UCT/CPU) qui exécute les programmes chargés en mémoire ;
- Les unités d'entrée/sortie qui permettent le lien et l'échange d'information avec les périphériques (clavier, écran, souris, imprimante, etc.).



# ARPANET (1969)

- ARPANET, qui finit par devenir Internet, commencé en connectant seulement 4 noeuds
- « Le ministère de la Défense » sponsorise l'ARPA ([Agence de projets de recherche avancée](#)) pour consolider d'un réseau interconnecté d'ordinateurs géographiquement éloignés

29 Oct 69	2100	LOADED	OP. PROGRAM ISK
		FOR BEN BARKER	
		B&V	
	22:30	TALKED TO SRI	CSE
		Host to Host	
		LEFT TOP IMP PROGRAM CSE	
		running after sending	
		a host dead message	
		to imp.	

Figure – Journal ARPANET IMP montrant que Charles S. Kline a envoyé son premier message sur ARPANET le 29 octobre 1969

# Besoin d'inventer un système d'exploitation

- Dans les premiers jours de l'informatique (années 50), les ordinateurs étaient des machines complètement ouvertes
- Pour exécuter un programme Fortran (1954) :

1. Réserver un créneau d'utilisation de l'ordinateur
2. Charger dans le lecteur de cartes perforées le bloc de cartes du compilateur (traducteur) Fortran
3. Charger les cartes du programme Fortran



## Besoin d'inventer un système d'exploitation

- Pour exécuter un programme Fortran (1954) : -suite-
  4. L'ordinateur lit les cartes ("lentement") parfois plusieurs fois
  5. Il produit des cartes avec le programme compilé (en langage machine) s'il n'y a pas d'erreurs
  6. Introduire les nouvelles cartes + les librairies
  7. S'il y a des erreurs, il faut corriger et répéter

# Besoin d'inventer un système d'exploitation

- Pour exécuter un programme Fortran (1954) : -suite-
  4. L'ordinateur lit les cartes ("lentement") parfois plusieurs fois
  5. Il produit des cartes avec le programme compilé (en langage machine) s'il n'y a pas d'erreurs
  6. Introduire les nouvelles cartes + les librairies
  7. S'il y a des erreurs, il faut corriger et répéter

⇒ Naissance (autour de 1960) des systèmes d'exploitation = programmes "permanents", qui évitent de répéter ces tâches

Un système d'exploitation fait bien plus de choses que cette prise en charge des programmes utilisateurs (objet du cours suivant)

## UNIX (1972)

- Système d'exploitation UNIX développé aux **Laboratoires Bell** par :
  - Ken Thompson
  - Dennis Ritchie
  - Brian Kernighan et d'autres
- Réécrit en 1972 utilisant langage C
- Plusieurs variantes d'UNIX sont populaires : **Mac OS X, Linux, BSD**



Figure – Ken Thompson



Figure – Dennis Ritchie



Figure – Brian Kernighan

# Microprocesseur (1971)

- Microprocesseur : la totalité du processeur (CPU) s'inscrit sur une seule puce
- **Trois sociétés** ont simultanément développé le microprocesseur de manière indépendante :
  - Texas Instruments
  - Intel
  - Garrett AiResearch

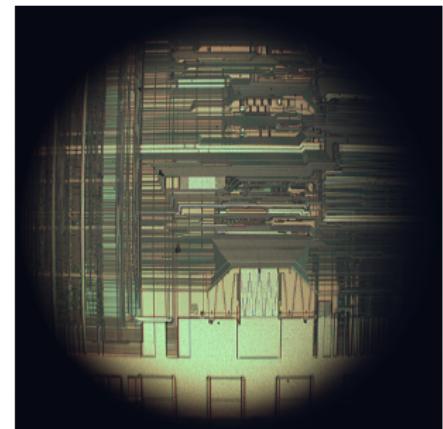
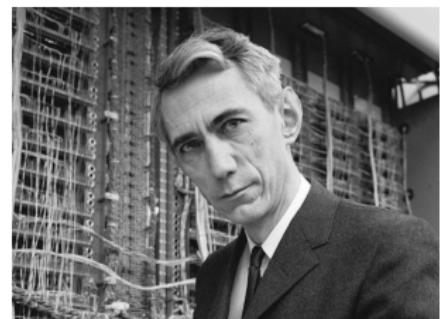


Figure – Regardez par un microscope (200x) sur un microprocesseur Intel

# Naissance de l'ordinateur

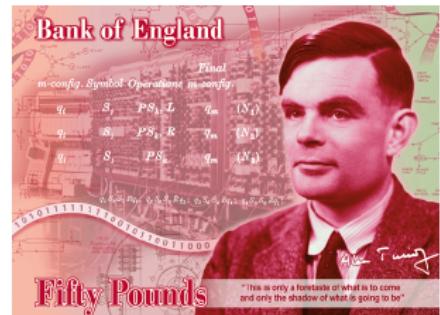
- Claude Shannon était un **mathématicien, ingénieur électronicien et cryptographe américain**, surnommé "le père de la théorie de l'information"
- Shannon est connu pour avoir fondé **la théorie de l'information** avec un article historique, "A Mathematical Theory of Communication", en **1948**
- Il démontre que **les applications électriques de l'algèbre booléenne pourraient construire toute relation numérique logique**  
⇒ Tout calcul peut être réalisé avec les 3 opérations logiques de base « et », « ou », et « non »



Figure

# Naissance de l'ordinateur (Alan Turing)

- Alan Turing (1936) : formalise une machine universelle ou Machine de Turing
- Décrivant un modèle abstrait du fonctionnement des appareils mécaniques de calcul
  - ⇒ La thèse de Turing postule que tout problème de calcul fondé sur une procédure algorithmique peut être résolu par une machine de Turing
  - ⇒ Invente le concept de programme.



Figure

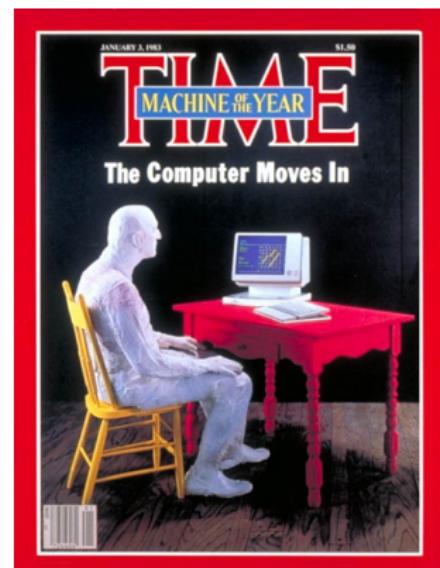
# Apple II Ordinateur Personnel (1977)

- Apple II, conçu principalement par **Steve Wozniak**
- Il est le premier ordinateur personnel produit en série et qui connaît un grand succès
- **Prix : 1300 \$ pour le modèle avec 4 Ko de RAM**
- **Prix : 2600 \$ pour un modèle de 48 Ko de RAM**



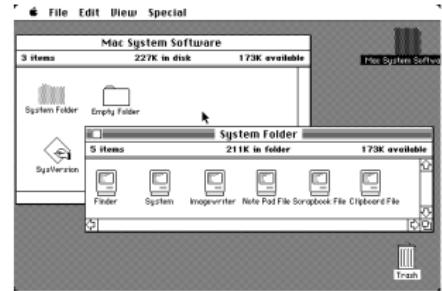
## IBM Ordinateur Personnel (1981)

- IBM développe un PC avec un microprocesseur Intel et le système d'exploitation DOS de Microsoft
- Prix commencé à 1565 \$
- 300 000 vendus en 1981
- 3 274 000 vendus en 1982



# Apple Macintosh (1984)

- Apple Computers (Steve Jobs et Steve Wozniak) ont développé le Macintosh qui a popularisé l'interface utilisateur graphique (GUI)
- La publicité commerciale d'Apple publiée en 1984 faisant la promotion du Macintosh
  - Cette publicité était la plus chère jamais produite à l'époque (environ 1 million de dollars)



# World Wide Web (WWW) (1990)

- Tim Berners-Lee du CERN développe le WWW
- Un réseau mondial de « documents interconnectés » sur Internet
- « Le Web » deviendra **populaire plusieurs années plus tard**, lorsque **Netscape** développée un « Web Navigateur »



# Microsoft Windows (1992)

- Microsoft (Bill Gates) lance Windows 3.1, la première version de Windows qui a connu un grand succès



## Deep Blue (1997)

- L'ordinateur **Deep Blue** d'IBM bat le champion du monde d'échecs **Garry Kasparov** lors de son **deuxième affrontement de six matchs**
- Devenant ainsi **le premier système informatique à battre un champion du monde** sous les contrôles de temps « de tournoi d'échecs standard »



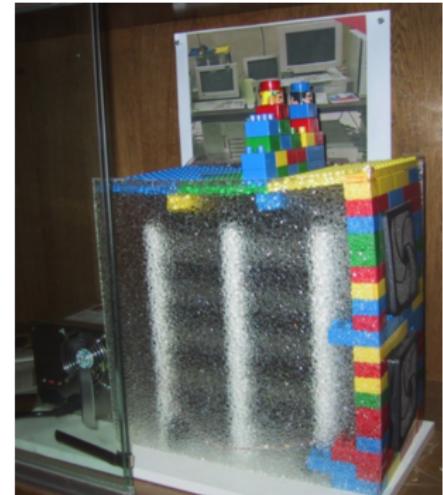
IBM's Deep Blue



World Champion Garry Kasparov

## Google (1998)

- Les doctorats : **Larry Page et Sergey Brin** de **Stanford** ont créé **Google**
- Un moteur de recherche Web qui utilise leur nouvel algorithme **PageRank** pour classer les résultats des moteurs de recherche
- Google provient d'une **faute d'orthographe** de Google qui est 1 suivi de 100 zéros



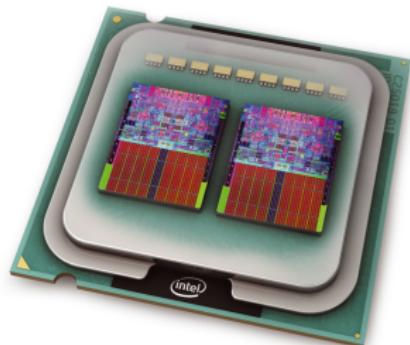
## Réseaux sociaux en ligne (200x)

- Les réseaux sociaux en ligne (et le partage d'informations) sont devenus populaires au début des années 2000
- 2002 : **Friendster** créé par **Jonathan Abrams et Peter Chin**
- 2003 : **MySpace** créé par les employés de **eUniverse**
- 2004 : **Facebook** créé par **Mark Zuckerberg** alors qu'il était étudiant à **Harvard**
- 2006 : **Twitter** créé par **Jack Dorsey**



## Processeurs multicoeurs (2005)

- Les PC avec les processeurs double cœur arrivent sur le marché.
- Les processeurs multicoeurs ont plusieurs processeurs sur une seule puce (chip)
- Ils permettent plus de débit avec une vitesse de processeur inférieure
- Utilisant ainsi moins d'énergie
- Met davantage l'accent sur la programmation parallèle



## iPhone (2007)

- iPhone par Apple révolutionne les interfaces à écran tactile pour les appareils mobiles



## iPad (2010)

- L'iPad d'Apple relance le marché de l'informatique sur tablette



## Voiture sans conducteur de Google (2012)

- Google obtient la première permis de conduire autonome dans le Nevada

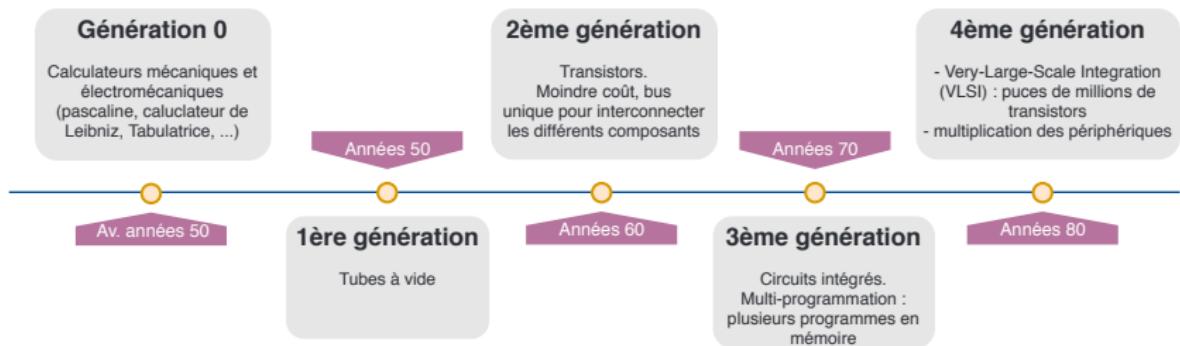


## Lunettes Google (2013)

- Google Glass affiche les informations dans un format mains libres, similaire à celui d'un smartphone
- Les utilisateurs ont communiqué avec Internet via des commandes vocales en langage naturel



# Naissance de l'industrie informatique



# Générations informatiques

Il y a cinq générations d'ordinateur :

- Première génération - 1946 -1958
- Deuxième génération - 1959 -1964
- Troisième génération - 1965-1970
- quatrième génération - de 1971 à aujourd'hui
- Cinquième génération - Aujourd'hui à l'avenir

# Première Génération

- Tubes à vide usagés pour circuits et tambours magnétiques pour mémoire
- Étaient souvent énormes, occupant des pièces entières.
- Étaient très coûteux à exploiter
- Utilisez beaucoup d'électricité
- Généré beaucoup de chaleur
- Ce qui était souvent la cause de dysfonctionnements



# Première Génération

- Basé sur le langage machine
- Le langage de programmation de niveau le plus bas pour effectuer des opérations
- Ne pouvait résoudre qu'un problème à la fois
- L'entrée était basée sur des cartes perforées et du ruban adhésif
- La sortie était affichée sur les impressions



## Deuxième Génération

- Les transistors ont remplacé les tubes à vide
- Un transistor a remplacé l'équivalent de 40 tubes à vide.
- Les ordinateurs deviennent plus petits, plus rapides, moins chers, plus écoénergétiques et plus fiables
- Génère encore beaucoup de chaleur qui peut endommager l'ordinateur
- Déplacé d'un langage machine binaire cryptique à un langage symbolique, ou d'assemblage
- Qui permettait aux programmeurs de spécifier des instructions en mot



## Deuxième Génération

- Toujours basé sur des cartes perforées pour l'entrée et des impressions pour la sortie
- Les premiers ordinateurs qui ont stocké leurs instructions dans leur mémoire,
- Qui est passé du tambour magnétique à la technologie du noyau magnétique.



# Troisième Génération

- Le développement du circuit intégré
- Les transistors ont été miniaturisés et placés sur des puces de silicium, appelées semi-conducteurs,
- Qui a considérablement augmenté la vitesse et l'efficacité des ordinateurs.
- Beaucoup plus petit et moins cher
- Peut exécuter des instructions en milliardièmes de seconde



## Troisième Génération

- Interaction à travers les claviers et les moniteurs et interfacée avec un système d'exploitation,
- Autorisé le périphérique à exécuter plusieurs applications différentes en même temps avec un programme central qui surveillait la mémoire
- Devenu accessible à un public de masse parce qu'ils étaient plus petits et moins chers que leurs prédecesseurs



## Quatrième Génération

- Le microprocesseur a été inventé
- Des milliers de circuits intégrés ont été construits sur une seule puce de silicium
- Les ordinateurs sont devenus plus puissants
- Pourraient être reliés entre eux pour former des réseaux, ont conduit au développement de l'Internet.
- Vu le développement des interfaces graphiques, de la souris et des appareils de poche

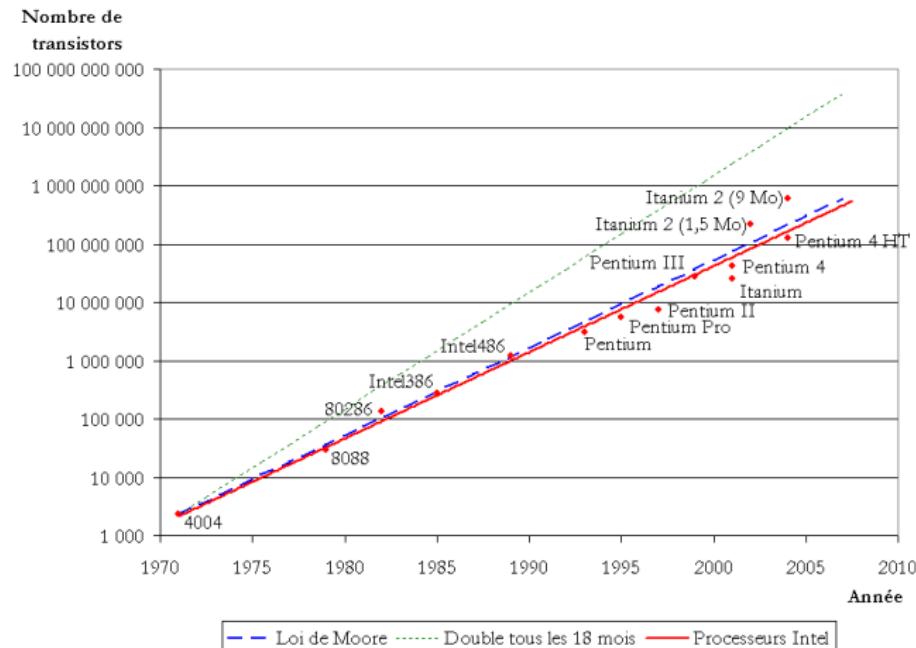


## Cinquième Génération

- Basé sur l'intelligence artificielle
- Encore en développement
- L'utilisation du traitement parallèle et des supraconducteurs contribue à faire de l'intelligence artificielle une réalité
- L'objectif est de développer des dispositifs qui répondent aux idées du langage naturel et qui sont capables d'apprentissage et d'auto-organisation.
- Certaines applications, telles que la reconnaissance vocale, sont utilisées aujourd'hui

## Loi de Moore (1965)

Nombre de transistors (des microprocesseurs = ~CPU) sur une puce de silicium double tous les deux ans.



# Évolution des microprocesseurs Intel

Date	Nom	Nombre de transistors	Finesse de gravure (µm)	Fréquence de l'horloge	Largeur des données	MIPS
1971	4004	2 300		108 kHz	4 bits/4 bits bus	
1974	8080	6 000	6	2 MHz	8 bits/8 bits bus	0,64
1979	8088	29 000	3	5 MHz	16 bits/8 bits bus	0,33
1982	80286	134 000	1,5	6 à 16 MHz (20 MHz chez AMD)	16 bits/16 bits bus	1
1985	80386	275 000	1,5	16 à 40 MHz	32 bits/32 bits bus	5
1989	80486	1 200 000	1	16 à 100 MHz	32 bits/32 bits bus	20
1993	Pentium	3 100 000	0,8 à 0,28	60 à 233 MHz	32 bits/64 bits bus	100
1997	Pentium II	7 500 000	0,35 à 0,25	233 à 450 MHz	32 bits/64 bits bus	300
1999	Pentium III	9 500 000	0,25 à 0,13	450 à 1 400 MHz	32 bits/64 bits bus	510
2000	Pentium 4	42 000 000	0,18 à 0,065	1,3 à 3,8 GHz	32 bits/64 bits bus	1 700
2004	Pentium 4D « Prescott »	125 000 000	0,09 à 0,065	2,66 à 3,6 GHz	32 bits/64 bits bus	9 000
2006	Core 2™ Duo	291 000 000	0,065	2,4 GHz (E6600)	64 bits/64 bits bus	22 000
2007	Core 2™ Quad	2*291 000 000	0,065	3 GHz (Q6850)	64 bits/64 bits bus	2*22 000 (?)
2008	Core 2™ Duo (Penryn)	410 000 000	0,045	3,33 GHz (E8600)	64 bits/64 bits bus	~24 200
2008	Core 2™ Quad (Penryn)	2*410 000 000	0,045	3,2 GHz (QX9770)	64 bits/64 bits bus	~2*24 200
2008	Intel Core i7 (Nehalem)	731 000 000	0,045 (2008) 0,032 (2009)	2,66 GHz (Core i7 920) 3,33 GHz (Core i7 Ext. Ed. 975)	64 bits/64 bits bus	?
2009	Intel Core i5/i7 (Lynnfield)	774 000 000	0,045 (2009)	2,66 GHz (Core i5 750) 2,93 GHz (Core i7 870)	64 bits/64 bits bus	?
2010	Intel Core i7 (Gulftown)	1 170 000 000	0,032	3,33 GHz (Core i7 980X)	64 bits/64 bits bus	?

## Programmes exécutés par les ordinateurs

- Programme = séquence d'instructions ;
- Instruction = additionner deux nombres, tester si un nombre est nul, déplacer un nombre vers/depuis la mémoire centrale, ... ;
- **Jeu d'instructions** que l'ordinateur sait exécuter forment un langage = **langage machine** ;
- Un fabricant de processeur décide du langage machine à fournir (**le plus simple possible pour réduire la complexité et le coût de l'électronique**)

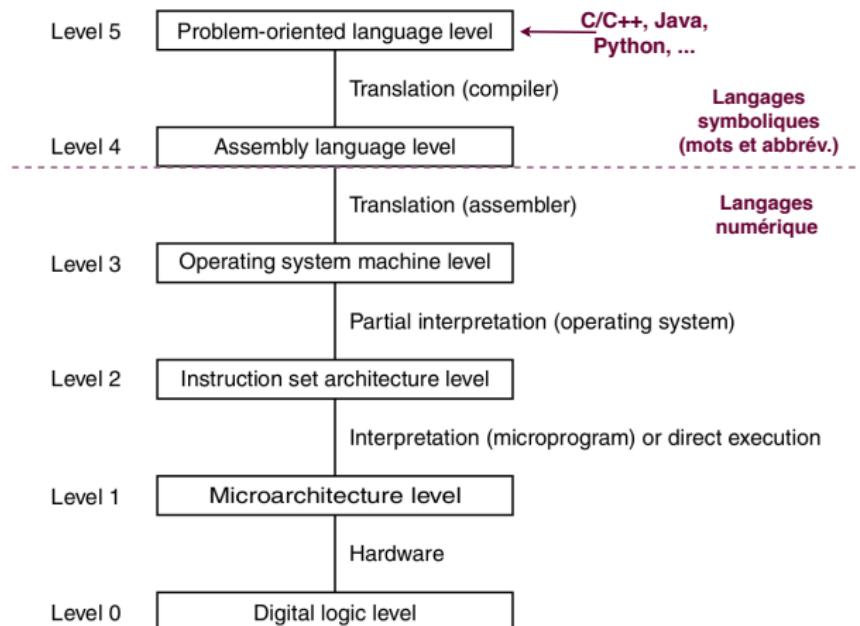
# Organisation structurée des ordinateurs

- Langage machine est souvent trop simple  
⇒ difficile à utiliser par les informaticiens **pour résoudre des problèmes de plus haut niveau** (que de simples additions, ...);
- L'ordinateur propose un langage (machine) L0 et l'utilisateur veut écrire un programme dans un langage plus évolué L1.
- Comme exécuter ces programmes écrits en L1?
  - les **traduire** vers L0 ;
  - écrire un programme dans L0, qui prend le programme écrit en L1 comme donnée et exécute, pour chaque instruction L1, des instructions L0 (un **interprète** ou **machine virtuelle**)

## Organisation en couches d'abstraction

- L0 et L1 ne doivent pas être “très” différents pour que la traduction ou l’interprétation soient efficaces et pas trop complexes ;

# Un ordi. avec une archi. multi-couches possible



# Remerciements et références

## Remerciements

Chouki Tibermacine, professeur à la FDS (mon prédécesseur)

## Références bibliographiques

- Andrew Tanenbaum. Architecture de l'ordinateur, 5ème édition. Pearson Education, 2009  
Andrew Tanenbaum. *Structured Computer Organization, 6th edition.* Pearson, 2012.
- Andrew Tanenbaum. Systèmes d'exploitation 3ème Ed. Pearson, 2008.  
Andrew Tanenbaum & Herbert Bos. *Modern Operating Systems.* Pearson 2016.